



**PURWARUPA PERAHU UNTUK MONITORING DAN  
KLASIFIKASI KUALITAS AIR BENDUNGAN DENGAN METODE  
K-NEAREST NEIGHBOR (KNN)**

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Teknik

Disusun oleh:

Chikam Muhammad

NIM: 165150300111041



PROGRAM STUDI TEKNIK KOMPUTER  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA

MALANG  
2020

## PENGESAHAN

PURWARUPA PERAHU UNTUK *MONITORING* DAN *KLASIFIKASI* KUALITAS AIR  
BENDUNGAN DENGAN METODE *K-NEAREST NEIGHBOR* (KNN)

SKRIPSI

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Teknik

Disusun Oleh :  
Chikam Muhammad  
NIM : 165150300111041

Skripsi ini telah diuji dan dinyatakan lulus pada  
24 April 2020

Telah diperiksa dan disetujui oleh :

Dosen Pembimbing I



Rizal Maulana, S.T., M.T., M.Sc.  
NIK: 201607 891009 1 001

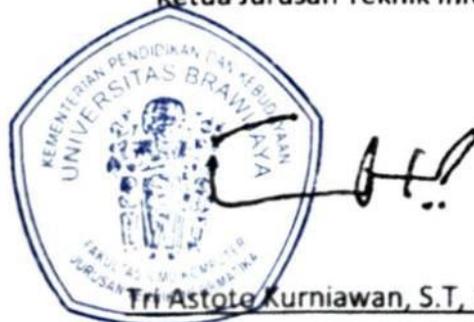
Dosen Pembimbing 2



Mochammad Hanhats Hanafi Ichsan, S.ST, M.T.  
NIP: 19881229 201903 1 010

Mengetahui

Ketua Jurusan Teknik Informatika



Tri Astote Kurniawan, S.T, M.T, Ph.D.  
NIP: 19710518 200312 1 001



## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang tertulis dalam dan disebutkan dalam daftar pustaka pada naskah ini.

Apabila ternyata di dalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 Ayat 2 dan Pasal 70).

Malang, 24 April 2020



Chikam Muhammad

NIM : 165150300111041



## ABSTRAK

**Chikam Muhammad, Purwarupa Perahu Untuk *Monitoring* dan Klasifikasi Kualitas Air Bendungan dengan Metode *K-NEAREST NEIGHTBOR* (k-NN).**

**Pembimbing : Rizal Maulana S.T., M.T., M.Sc dan Mochammad Hannats Hanafi Ichsan, S.T.,M.T.**

Bendungan air memiliki banyak fungsi seperti sebagai penampung air, sebagai tempat PLTA, sebagai pengendali banjir serta dapat digunakan sebagai wahana rekreasi dan pariwisata. Selama ini kegiatan *monitoring* kualitas air bendungan yang dilakukan oleh pihak terkait masih dilakukan secara manual. Pihak terkait perlu melakukan *monitoring* kualitas air pada *inlet* (titik masuk air bendungan), titik tengah bendungan dan *outlet* (titik keluar air pada bendungan) sehingga hal tersebut masih memakan waktu yang cukup bagi pihak terkait. Dari permasalahan tersebut maka penelitian ini melakukan inovasi dengan membuat sistem yang bisa bergerak dengan 5 parameter kualitas air untuk mengukur kualitas air pada bendungan. Sistem yang dibuat menggunakan 2 *microcontroller* yaitu arduino nano untuk melakukan akuisisi data dan node MCU untuk melakukan proses klasifikasi k-NN serta mengirim data ke *smartphone*. Parameter yang diukur oleh purwarupa alat yang dibuat yaitu pH air, jumlah zat yang terlarut (TDS), suhu air, kekeruhan air dan kedalaman air. Selain itu sistem yang dibuat menambahkan metode k-NN untuk melakukan klasifikasi kualitas air tersebut. Hasil kelas kualitas air dari klasifikasi k-NN yaitu kelas “Bagus”, “Sedang”, “Jelek”. Purwarupa alat yang dibuat pada penelitian ini dapat bergerak ke seluruh area bendungan air untuk melakukan *monitoring* dan dapat mengirimkan data kualitas air ke *smartphone*. Dengan data yang dapat diakses pada *smartphone* akan mempermudah pihak terkait untuk melihat kualitas air pada bendungan air secara keseluruhan dengan purwarupa alat yang bergerak sesuai perintah. Klasifikasi k-NN menggunakan 50 data latih dan 25 data uji untuk menguji metode yang digunakan. Hasil pengujian klasifikasi k-NN didapatkan akurasi sebesar 92%. Pengujian waktu komputasi dari sistem yang dibuat dilakukan sebanyak 10 kali pengujian dan didapatkan hasil rata-rata waktu komputasi sebesar 4135 ms.

**Kata Kunci : *Monitoring*, Kualitas Air, Bendungan, k-NN.**



## ABSTRACT

**Chikam Muhammad, Boat Prototype For Monitoring and Classification Dam Water Quality with K-Nearest Neighbor Method (k-NN).**

**Supervisor : Rizal Maulana S.T., M.T., M.Sc and Mochammad Hannats Hanafi Ichsan S.T., M.T.**

Water dams have many functions, such as water resevoirs, PLTA sites, flood control and can be used for recreation and tourism. Water quality monitoring activities that most people do nowadays still using manual method and less effective. For dam monitoring activities, related parties need to manually monitor water quality at dams water inlet, midpoint and water outlet. From these problem, this research makes innovation by creating a system than can move with five parameters of water quality. This research uses two microcontrollers for procesing purposes. Arduino nano is used for data acquisition while node MCU is used for k-NN classification method and sending five sensors data along with the classification result to firebase. so it can be displayed to user applications. The parameters measured by the boat are pH water, water temperature, total dissolved solid in water, water turbidity and water depth. In addition, the system is made by using k-NN method for water quality classification purpose. The results of the classification are three classes consisting of "Good", "Medium", "Bad" class. The system made in this research can move to the entire dam area to monitor water quality and able to send water quality data to smartphone. With data that can be accessed on the smartphone, this will make easier for related parties to look for the water quality in the water dam as a whole with a boat that moves according to orders. The k-NN classification uses 50 training data and 25 test data. The result obtained from classification test is 92% of accuracy. The computanional time of the system measured with 10 testing and the results obtained with an average of 4135 ms of computing time.

**Keywords : Monitoring, Water Quality, Dams Water, k-NN**



## DAFTAR ISI

PENGESAHAN.....	ii
PERNYATAAN ORISINALITAS.....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xiii
DAFTAR LAMPIRAN.....	xv
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan.....	3
1.4 Manfaat.....	4
1.5 Batasan Masalah.....	4
1.6 Sistematika Pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN.....	6
2.1 Isi Landasan Kepustakaan.....	6
2.1.1 <i>Implementation of Internet of Things for Water Quality</i> .....	7
2.1.2 Implementasi Metode Fuzzy Pada Kualitas Air Kolam Bibit Lele Berdasarkan Suhu dan Kekeruhan.....	7
2.1.3 Analisis Kualitas Air Dan Strategi Pengendalian Pencemaran Air Sungai Metro di Kota Kepanjen Kabupaten Malang.....	8
2.1.4 <i>Real-Time Water Quality Monitoring System Using Internet of Things</i> .....	8
2.1.5 Implementasi Algoritme <i>K-Nearest Neighbor</i> Untuk Identifikasi Kualitas Air (Studi Kasus: Pdam Kota Surakarta).....	9
2.2 Dasar Teori.....	9
2.2.1 Karakteristik Air.....	10
2.2.2 Bendungan Air.....	11
2.2.3 Sensor Kedalaman Air.....	12
2.2.4 Sensor pH air.....	13









## DAFTAR TABEL

Tabel 2.1 Penelitian Terkait .....	6
Tabel 4.1 Kebutuhan Fungsional Sistem .....	28
Tabel 4.2 Tabel Spesifikasi Arduino Nano .....	32
Tabel 4.3 Spesifikasi Tabel Node MCU .....	32
Tabel 4.4 Spesifikasi Sensor TDS .....	33
Tabel 4.5 Spesifikasi Sensor Kedalaman Air .....	33
Tabel 4.6 Tabel Spesifikasi Sensor Kekeruhan Air .....	34
Tabel 4.7 Spesifikasi Sensor Suhu .....	34
Tabel 4.8 Spesifikasi Sensor pH Air .....	35
Tabel 4.9 Spesifikasi Motor DC 12V .....	35
Tabel 4.10 Spesifikasi <i>Driver</i> Motor L298N .....	35
Tabel 5.1 Skema Pin Arduino Nano .....	39
Tabel 5.2 Skema Pin Node MCU .....	41
Tabel 5.3 Tabel Skema Pin Sensor pH dengan Arduino Nano .....	41
Tabel 5.4 Tabel Skema Pin Sensor pH dengan <i>Driver</i> Motor .....	42
Tabel 5.5 Tabel Skema Pin Sensor Suhu DS18B20 dengan Arduino Nano .....	42
Tabel 5.6 Tabel Skema Pin Sensor Suhu DS18B20 dengan <i>Driver</i> Motor .....	43
Tabel 5.7 Tabel Skema Pin Sensor JSN-SR04T dengan Arduino Nano .....	43
Tabel 5.8 Tabel Skema Pin Sensor JSN-SR04T dengan <i>Driver</i> Motor .....	44
Tabel 5.9 Tabel Skema Pin Sensor <i>Turbidity</i> dengan Arduino Nano .....	44
Tabel 5.10 Tabel Skema Pin Sensor <i>Turbidity</i> dengan <i>Driver</i> Motor .....	44
Tabel 5.11 Tabel Skema Pin Sensor TDS dengan Arduino Nano .....	45
Tabel 5.12 Tabel Skema Pin Sensor TDS dengan <i>Driver</i> Motor .....	45
Tabel 5.13 Baris Program Akuisisi Sensor pH .....	65
Tabel 5.14 Baris Program Akuisisi Sensor Suhu .....	66
Tabel 5.15 Baris Program Akuisisi Sensor Kedalaman .....	67
Tabel 5.16 Baris Program Akuisisi Sensor Kekeruhan .....	68
Tabel 5.17 Baris Program Akuisisi Sensor TDS .....	68
Tabel 5.18 Baris Program Mengirim Data Arduino Nano ke Node MCU .....	70
Tabel 5.19 Baris Program Menerima Data Node MCU dari Arduino Nano .....	70
Tabel 5.20 Baris Program Klasifikasi k-NN .....	71



**DAFTAR GAMBAR**

Gambar 2.1 Gambar Bendungan Air .....	12
Gambar 2.2 Sensor JSN-SR04T .....	12
Gambar 2.3 Sensor pH .....	13
Gambar 2.4 Sensor Suhu DS18B20 .....	13
Gambar 2.5 Sensor <i>Turbidity</i> .....	14
Gambar 2.6 Sensor TDS .....	14
Gambar 2.7 Arduino Nano .....	15
Gambar 2.8 Node MCU .....	15
Gambar 2.9 <i>Firebase</i> .....	16
Gambar 2.10 Tampilan awal pada Arduino IDE .....	16
Gambar 2.11 Android Studio .....	17
Gambar 3.1 Titik-titik Pengumpulan Data .....	20
Gambar 3.2 Diagram Alir Pengerjaan Penelitian .....	21
Gambar 3.3 Blok Diagram Perancangan Sistem Perangkat Keras .....	24
Gambar 3.4 Blok Sistem Perancangan Perangkat Lunak .....	25
Gambar 5.1 Perancangan Purwarupa Alat Untuk <i>Monitoring</i> .....	38
Gambar 5.2 Perancangan Perangkat Keras Keseluruhan Sistem .....	40
Gambar 5.3 Perancangan Skema Sensor pH-014 .....	42
Gambar 5.4 Perancangan Skema Sensor Suhu DS18B20 .....	43
Gambar 5.5 Perancangan Skema Sensor Kedalaman JSN-SR04T .....	44
Gambar 5.6 Perancangan Skema Sensor Kekeruhan <i>Turbidity</i> .....	45
Gambar 5.7 Perancangan Skema Sensor TDS .....	46
Gambar 5.8 Perancangan Keseluruhan Sistem .....	47
Gambar 5.9 Perancangan Fungsi Deteksi Sensor pH-014 .....	48
Gambar 5.10 Perancangan Fungsi Deteksi Sensor Suhu Air .....	48
Gambar 5.11 Perancangan Fungsi Deteksi Sensor Kedalaman Air .....	49
Gambar 5.12 Perancangan Fungsi Deteksi Sensor Kekeruhan Air .....	50
Gambar 5.13 Perancangan Fungsi Deteksi Sensor TDS .....	51
Gambar 5.14 Perancangan Pengiriman Data Arduino ke Node MCU .....	52
Gambar 5.15 Perancangan Proses Menerima Data Node MCU dari Arduino .....	52
Gambar 5.16 Perancangan Klasifikasi k-NN .....	54



Gambar 5.17 Perancangan Pengiriman data Node MCU .....	55
Gambar 5.18 Perancangan Penerimaan Data dari Aplikasi .....	56
Gambar 5.19 Perancangan Pergerakan Purwarupa Alat <i>Monitoring</i> .....	57
Gambar 5.20 Sitemap Perancangan Aplikasi .....	58
Gambar 5.21 Perancangan Desain Aplikasi Halaman Awal <i>Smartphone</i> .....	59
Gambar 5.22 Perancangan Tampilan Isi Aplikasi .....	59
Gambar 5.23 Implementasi Purwarupa Alat Untuk <i>Monitoring</i> .....	61
Gambar 5.24 Implementasi Skema Sensor pH-014 .....	62
Gambar 5.25 Implementasi Skema Sensor Suhu DS18B20 .....	63
Gambar 5.26 Implementasi Skema Sensor JSN-SR04T .....	63
Gambar 5.27 Implementasi Skema Sensor <i>Turbidity</i> .....	64
Gambar 5.28 Implementasi Skema Sensor TDS .....	64
Gambar 5.29 Implementasi Desain Aplikasi Halaman Awal <i>Smartphone</i> .....	77
Gambar 5.30 Implementasi Tampilan isi Aplikasi .....	77
Gambar 6.1 Lokasi Pengujian Alat di Desa Kresek .....	80
Gambar 6.2 Lokasi Pengujian Alat di Desa Golang .....	81
Gambar 6.3 Lokasi Pengujian Alat di Desa Giringan .....	81
Gambar 6.4 Pengujian Mengakuisisi Data Sensor pH .....	83
Gambar 6.5 Hasil Serial Monitor Pengujian Sensor pH-014 .....	84
Gambar 6.6 Pengujian Mengakuisisi Data Sensor Suhu .....	85
Gambar 6.7 Hasil Serial Monitor Pengujian Sensor Suhu .....	86
Gambar 6.8 Hasil Serial Monitor Pengujian Sensor Kedalaman Air .....	89
Gambar 6.9 Hasil Pengujian Mengakuisisi Hasil Data Sensor <i>Turbidity</i> .....	91
Gambar 6.10 Hasil Serial Monitor Pengujian Sensor Kekeruhan Air .....	91
Gambar 6.11 Pengujian Akuisisi Hasil Data Sensor TDS .....	93
Gambar 6.12 Hasil Serial Monitor Pengujian Sensor TDS .....	94
Gambar 6.13 Penggunaan <i>Memory</i> pada Arduino Nano .....	104
Gambar 6.14 Penggunaan <i>Memory</i> pada Node MCU .....	105
Gambar 6.15 Pengujian Waktu Komputasi Arduino Nano .....	105
Gambar 6.16 Pengujian Waktu Komputasi Node MCU .....	105
Gambar 6.17 Pengujian <i>Reliabilitas</i> Proses Pengiriman Data Ke <i>Firestore Database</i> .....	108
Gambar 6.18 Gambar Pengujian Keseluruhan Sistem Gerak Purwarupa Alat .....	109



**DAFTAR LAMPIRAN**

LAMPIRAN A – DATA LATIH ..... 114

LAMPIRAN B – DATA UJI ..... 116

LAMPIRAN C – KODE PROGRAM ARDUINO NANO ..... 117

LAMPIRAN D – KODE PROGRAM NODE MCU ..... 118

LAMPIRAN E – PP 20 TAHUN 1990 TENTANG PENGENDALIAN PENCEMARAN AIR  
..... 123



## BAB 1 PENDAHULUAN

### 1.1 Latar Belakang

Air menjadi kebutuhan utama bagi kehidupan di bumi, sehingga tidak akan ada kehidupan seandainya jika tidak ada air di bumi. Dengan sangat pentingnya kehidupan terdapat air, maka perlu adanya upaya manusia dalam melakukan perlindungan terhadap sumber air (Effendi, 2003). Kualitas air dapat dinyatakan dengan beberapa parameter seperti parameter fisika (kekeruhan, zat padat terlarut, suhu dan sebagainya), parameter kimia (pH, oksigen terlarut, BOD (*Biological oxygen demand*), kadar logam dan sebagainya), dan parameter biologi (keberadaan plankton, organisme *macrobenthic*, energi, dan sebagainya). Peraturan Pemerintah No.20 tahun 1990 mengelompokkan kualitas air menjadi beberapa golongan yaitu Golongan A untuk air minum tanpa diolah terlebih dahulu, Golongan B sebagai bahan baku air minum, Golongan C untuk keperluan perikanan dan peternakan dan Golongan D untuk keperluan pertanian, industri dan pembangkit listrik tenaga air. Proses pengukuran kualitas air memiliki kerumitan sendiri (Ichsan, 2016). Objek lokasi pengukuran kualitas air yang dilakukan oleh pihak terkait biasanya seperti di lokasi sungai, bendungan dan waduk.

Bendungan atau bendungan sendiri merupakan sebuah bangunan dengan cekungan yang berisi air dan dibuat secara sengaja oleh pemerintah atau terbentuk secara alami. Peraturan pemerintah no 37 tahun 2010 menjelaskan bahwa bendungan dan waduk sendiri memiliki fungsi yang sama seperti sebagai penampung air, sebagai irigasi pertanian dan perkebunan, sebagai sumber air untuk kolam perikanan masyarakat dan sebagai sumber air bagi masyarakat sekitar ketika musim kemarau yang berkepanjangan. Waduk dan bendungan hanya memiliki perbedaan yaitu luas wilayah cekungan airnya, jika waduk memiliki luas yang lebih besar ketimbang bendungan. Dengan banyaknya manfaat dari bendungan dan waduk maka air yang ada pada bendungan dan waduk perlu dijaga dan di monitoring kualitas air nya agar kualitas air yang ada pada bendungan dan waduk dapat terjaga dengan baik. Proses monitoring kualitas air yang ada bendungan dan waduk sangat diperlukan karena ketika air yang ada pada bendungan dan waduk memiliki kualitas yang jelek akan mengakibatkan banyak masalah. Masalah tersebut seperti kolam ikan masyarakat yang menggunakan airnya dari bendungan atau waduk akan mengalami permasalahan. Selain itu contoh masalahnya yaitu ketika masyarakat memiliki hewan ternak yang kebutuhan air minunya dari air bendungan dan waduk maka hewan ternak tersebut akan memiliki permasalahan. Selama ini *monitoring* kualitas air di waduk dan bendungan masih dilakukan secara manual, maka diperlukan alat untuk melakukan *monitoring* kualitas air secara otomatis.

Sudah ada penelitian saat ini yang berjudul "*Real-Time Water Quality Monitoring System Using Internet of Things*". Pada penelitian ini menggunakan empat buah parameter yaitu kekeruhan air,  $CO_2$ , pH air dan suhu air. Penelitian yang dikerjakan oleh mahasiswa di India yaitu Brinda Das dan P.C. Jain lebih



mengedepankan tentang *monitoring* kualitas air secara *real time*. Untuk tempat lokasi penelitian ini terletak di sungai gangga India. Penelitian tersebut menggunakan LCD 16x2 sebagai *output* dari sistem dan *database* ThingSpeak agar data hasil *monitoring* biasa diakses melalui web dan *smartphone*. Kekurangan dari penelitian ini belum adanya kecerdasan buatan yang digunakan mengklasifikasikan data yang diterima untuk menentukan kualitas air. Selain itu penelitian ini menggunakan purwarupa yang hanya bisa melakukan *monitoring* secara manual di suatu tempat saja tanpa bisa berpindah pindah secara otomatis (Brinda dan Jain, 2017).

Adanya kekurangan pada penelitian yang sudah dibuat di atas, maka peneliti melakukan penelitian dengan membuat alat yang bisa digunakan *monitoring* air bendungan dengan menggunakan parameter tingkat pH, kedalaman air, suhu, kekeruhan air, serta nilai kandungan jumlah zat yang terlarut (*Total Dissolved Solids*). Alat pada penelitian ini akan dibuat yang dilengkapi motor sehingga alat tersebut bisa bergerak sesuai dengan perintah yang diharapkan bisa melakukan akuisisi data tentang kualitas air di sebuah waduk secara merata dan secara otomatis. Dengan kelima parameter yang diukur diharapkan akan menggambarkan kualitas air yang sebenarnya. Data yang didapat dari sensor akan diproses arduino nano sebagai *microcontroller*. Pada alat yang dibuat dilengkapi node MCU yang digunakan untuk melakukan klasifikasi k-NN terhadap kualitas air dan mengirim data ke *database*, sehingga hasil klasifikasi bisa diakses oleh *smartphone* pengguna. Dengan penggunaan *smartphone* untuk melihat proses klasifikasi, akan mempermudah masyarakat dalam melakukan *monitoring* kualitas air.

Metode k-NN (*k-Nearest Neighbor*) adalah suatu algoritme yang memiliki tujuan untuk mengklasifikasikan suatu objek baru berdasarkan data *training*. Metode k-NN sendiri termasuk *supervised learning* yang memerlukan data latih terlebih dahulu (Han, 2006), sehingga hasilnya yang diolah akan diklasifikasikan berdasarkan jarak tetangga terdekat. Terdapat 2 cara *voting* untuk menentukan proses klasifikasi dari metode k-NN. 2 cara *voting* tersebut adalah *voting weighted* dan *voting unweighted*. Dari 2 cara *voting* tersebut terdapat kelebihan dan kelemahan antara masing-masing cara *voting* tersebut. Dengan metode k-NN akan membuat sistem lebih akurat dalam melakukan klasifikasi kualitas air.

Beberapa permasalahan seperti di atas bisa disimpulkan untuk merancang suatu sistem yang memiliki lima buah parameter yang digunakan untuk mengukur kualitas air. Parameter kualitas air menggunakan parameter tingkat pH, kedalaman air, suhu, kekeruhan air, serta nilai kandungan jumlah zat yang terlarut di air yang diklasifikasikan dengan menggunakan metode k-NN. Metode *K-Nearest Neighbor* dipilih karena memiliki akurasi yang tinggi dan rasio kesalahan yang kecil (Yolanda, 2018). Selain itu metode k-NN juga efektif dalam proses klasifikasi jika terdapat data latih yang berjumlah banyak, mudah dalam proses pelatihan dan tahan ketika terdapat data latih yang memiliki *noise* (Mutrofin, siti et al, 2014). Pada penelitian ini menggunakan cara *weighted voting* karena cara *weighted voting* lebih baik dan dapat memecahkan masalah yang tidak bisa dipecahkan oleh cara *unweighted voting* di beberapa kondisi tertentu. Lalu untuk alat yang dibuat menggunakan



sumber daya dari 1 buah baterai *Lippo* berkapasitas 2600 mAh. Baterai *Lippo* cocok digunakan karena dapat mencukupi kebutuhan dari keseluruhan sistem yang akan dibuat dan memiliki siklus isi ulang yang tahan lama sehingga memiliki umur penggunaan yang lebih lama (Maulana, 2018). Alat pada penelitian ini akan diletakkan di bagian tempat air masuk bendungan yang biasa disebut *inlet*, di tengah-tengah bendungan, dan di bagian air keluarnya bendungan yang biasanya disebut *outlet*. 3 titik yang akan di *monitoring* adalah titik vital di mana kualitas air bisa tergambar karena untuk tempat masuk air akan menjadi pengaruh terhadap kualitas air yang berada di tengah bendungan, di tengah bendungan akan menggambarkan kualitas air yang berada di bendungan tersebut dan tempat keluar air bendungan akan menggambarkan kualitas air yang keluar menuju sungai ataupun irigasi masyarakat (Rosada, 2019). Selain itu untuk menggambarkan kualitas air yang ada di bendungan tersebut maka alat yang dibuat akan melakukan *monitoring* di bendungan tersebut dengan beberapa titik di dalamnya. Titik-titik tersebut ditentukan dengan luas dari bendungan tersebut. Rentang antar titik untuk melakukan *monitoring* di bendungan  $\pm 15$  meter. Alat ini diperuntukkan untuk masyarakat, pemerintah dan pemilik keramba jaring apung. Alat di dalam penelitian ini diharapkan mempermudah masyarakat dan pemerintah dalam melihat kualitas air di sekitar dan melakukan *monitoring*.

## 1.2 Rumusan Masalah

1. Bagaimana implementasi alat untuk klasifikasi dan *monitoring* air bendungan berbasis arduino nano dan node MCU menggunakan metode k-NN?
2. Bagaimana akurasi dan hasil analisis dari sensor pH-014, sensor TDS, sensor *turbidity*, sensor kedalaman air JSN-SR04T, dan sensor suhu air DS18B20?
3. Bagaimana hasil analisis dan tingkat akurasi dalam klasifikasi air menggunakan metode k-NN?
4. Bagaimana waktu komputasi sistem dan penggunaan *memory* yang diperlukan dari sistem pada penelitian dengan klasifikasi metode k-NN?
5. Bagaimana tingkat *reliabilitas* proses pengiriman data yang dikirim sistem ke firebase *database*?

## 1.3 Tujuan

1. Untuk mengetahui implementasi alat untuk *monitoring* kualitas air berbasis arduino menggunakan metode k-NN menggunakan arduino nano dan node MCU.
2. Untuk mengetahui akurasi dari sensor pH, sensor TDS, sensor *turbidity*, sensor kedalaman air JSN-SR04T dan sensor suhu DS18B20.
3. Untuk menguji tingkat akurasi klasifikasi kualitas air bendungan menggunakan metode k-NN.



4. Untuk menguji waktu komputasi dan penggunaan *memory* yang diperlukan dari sistem dengan klasifikasi k-NN dengan data lima sensor yang digunakan.
5. Untuk menguji dan mengetahui tingkat *reliabilitas* proses pengiriman data yang dilakukan oleh sistem ke firebase *database*.

#### 1.4 Manfaat

Manfaat yang diperoleh dalam melakukan penelitian ini adalah sebagai berikut:

1. Dapat menjadi rujukan penelitian selanjutnya yang berhubungan dengan alat untuk *monitoring* kualitas air.
2. Dapat merancang alat yang bisa untuk *monitoring* kualitas air menggunakan *microcontroller* arduino nano dan node MCU.
3. Dapat menjadi rujukan penelitian selanjutnya yang berhubungan dengan penggunaan 2 *microcontroller* dalam sebuah sistem yang dibuat.

#### 1.5 Batasan Masalah

Agar dapat dilakukan secara lebih terfokus, maka penelitian ini dibatasi pada hal-hal berikut :

1. *Monitoring* dan klasifikasi air yang dilakukan dengan tempat di bendungan air dan waduk saja.
2. Perancangan alat ini dibuat dengan menggunakan purwarupa yang akan di tempatkan di waduk dan bendungan air.
3. Alat yang dibuat di peruntukan untuk bendungan air dan waduk dengan kecepatan air yang mengalir dengan batas maksimum 10 km/jam.
4. *Monitoring* dilakukan dengan beberapa titik di dalam bendungan dengan jarak antar bendungan sekitar  $\pm 15$  meter.

#### 1.6 Sistematika Pembahasan

Sistematik penulisan penelitian di tunjukkan untuk memberikan gambaran dan uraian dari penyusunan tugas akhir secara garis besar yang meliputi beberapa bab, sebagai berikut:

##### BAB 1 : Pedahuluan

Menguraikan latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan.

##### BAB 2 : Landasan Kepustakaan

Menguraikan kajian pustaka dan dasar teori yang mendasari klasifikasi dan *monitoring* air bedungan dengan metode k-NN dan sistem alat yang sudah ada.

**BAB 3 : Metodologi**

Menguraikan tentang metode dan langkah kerja yang terdiri dari studi literatur, analisis kebutuhan simulasi, perancangan sistem, implementasi dan analisis serta pengambilan kesimpulan.

**BAB 4 : Rekayasa Kebutuhan**

Menguraikan proses rekayasa kebutuhan untuk proses perancangan dan implementasi sistem.

**BAB 5 : Perancangan dan Implementasi**

Memuat proses perancangan sesuai rekayasa kebutuhan dan implementasi terhadap perancangan sistem yang telah direalisasikan sebelumnya.

**BAB 6 : Pengujian dan analisis**

Memuat hasil dan proses pengujian terhadap sensor yang ada dan klasifikasi kondisi air.

**BAB 7 : Kesimpulan**

Memuat kesimpulan yang diperoleh dari pembuatan dan pengujian program, serta saran-saran untuk mengembangkan lebih lanjut.

## BAB 2 LANDASAN KEPUSTAKAAN

Bab ini berisi pokok bahasan tinjauan pustaka yang mencakup kajian pustaka dan dasar teori sesuai dengan yang diperlukan. Kajian pustaka membahas beberapa penelitian yang telah dilakukan sebelumnya. Dasar teori berisi pokok bahasan yang diperlukan untuk menyusun penelitian yang dilakukan. Pada penelitian ini bersifat implementatif dengan bentuk pengembangan (*development*).

### 2.1 Isi Landasan Kepustakaan

Dalam pengerjaan penelitian penulis melakukan tinjauan pustaka. Berikut penelitian yang terkait dengan Purwarupa Perahu untuk *Monitoring* dan Klasifikasi Kualitas Air Bendungan Dengan Metode *k-Nearest Neighbor* (k-NN) seperti pada Tabel 2.1 berikut ini.

Tabel 2.1 Penelitian Terkait

No	Nama Penulis (Tahun), Judul Penelitian	Persamaan	Perbedaan	
			Penelitian Terdahulu	Rencana Penelitian
1	Priya S. Bhagat et al (2019) <i>Implementation of Internet of Things for Water Quality Monitoring.</i>	<i>Monitoring</i> kualitas air dan menggunakan node MCU sebagai mikrokontroller nya	Sistem <i>monitoring</i> kualitas air menggunakan <i>cloud Thinger.io</i> dan objek penelitiannya sungai.	Sistem <i>monitoring</i> kualitas air bendungan dan waduk dengan menggunakan metode k-NN dan menggunakan <i>Firestore database</i> .
2	Muchammad Cholilulloh (2018) Implementasi Metode <i>Fuzzy</i> Pada Kualitas Air Kolam Bibit Lele Berdasarkan Suhu dan Kekeruhan.	<i>Monitoring</i> kualitas air dengan indikator suhu dan kekeruhan air.	Objek yang diteliti untuk <i>monitoring</i> kualitas air adalah kolam lele.	Sistem yang dibuat digunakan untuk <i>monitoring</i> kualitas air bendungan dan waduk.
3	Mahyudin (2015) Analisis Kualitas Air Dan Strategi Pengendalian Pencemaran Air Sungai Metro di Kota	Melakukan <i>monitoring</i> dan analisis kualitas air.	Melakukan pengambilan sampel air dan dibawa ke laboratorium untuk proses	Melakukan <i>monitoring</i> dan Analisis kualitas air dilakukan secara langsung tanpa harus



	Kepanjen Kabupaten Malang		pengecekan kualitas air.	dibawa ke Lab terlebih dahulu.
4	Das Brinda, P.C. Jain (2017) <i>Real-Time Water Quality Monitoring System using Internet of Things</i> .	Data <i>monitoring</i> dapat diakses oleh pengguna melalui <i>smartphone</i> .	Belum adanya metode untuk klasifikasi kualitas air.	Menggunakan metode k-NN untuk klasifikasi hasil <i>monitoring</i> kualitas air.
5	Rio Adi Arnomo (2018) Implementasi Algoritme <i>K-Nearest Neighbor</i> Untuk Identifikasi Kualitas Air (Studi Kasus: Pdam Kota Surakarta).	Implementasi Algoritme <i>K-Nearest Neighbor</i> Untuk Identifikasi Kualitas Air	Data yang diperoleh dari Pdam Kota Surakarta dan parameter yang di uji berbeda.	Data diperoleh melalui akuisisi sensor yang digunakan pada bendungan dan menggunakan 5 parameter kualitas air.

### 2.1.1 Implementation of Internet of Things for Water Quality

Pada penelitian pertama ini yang akan dijadikan referensi dengan judul "*Implementation of Internet of Things for Water Quality Monitoring*". Pada penelitian ini menggunakan parameter suhu air, pH air dan CO<sub>2</sub> air sebagai parameter untuk menentukan klasifikasi kondisi air. Terdapat modul node MCU di sistem yang dibuat dengan tujuan untuk mengirim data ke *cloud Thinger.io* agar data yang ada dapat diakses. Selain itu pada penelitian ini menggunakan LCD 16x2 untuk mengetahui nilai dari tiap sensor ketika akuisisi data kualitas air. Kelebihan dari penelitian ini yaitu menggunakan gabungan dari beberapa metode yang membuat akurasi klasifikasi menjadi lebih baik. Kekurangan dari penelitian ini hanya menggunakan parameter suhu, pH dan CO<sub>2</sub> yang kurang bisa menggambarkan kondisi kualitas air yang sebenarnya. Selain itu penelitian ini belum adanya kecerdasan buatan yang bisa untuk mengklasifikasikan kualitas air tersebut. Pada penelitian yang pertama ini saya jadikan referensi pada bagian pengiriman data dan proses klasifikasi airnya untuk tiap sensor.

### 2.1.2 Implementasi Metode Fuzzy Pada Kualitas Air Kolam Bibit Lele Berdasarkan Suhu dan Kekeruhan

Pada penelitian kedua yang dijadikan referensi dengan judul "Implementasi Metode *Fuzzy* Pada Kualitas Air Kolam Bibit Lele Berdasarkan Suhu dan Kekeruhan". Pada penelitian ini berisi penelitian yang membahas sebuah sistem yang diciptakan untuk tujuan mengontrol suhu kolam lele dan kekeruhan airnya. Sistem yang dibuat memiliki dua buah *input* berupa sensor DS18B20 yang digunakan untuk mengukur suhu air kolam dan rangkaian LED dan LDR yang digunakan untuk mengukur tingkat



kekeruhan air kolam. Sistem yang dibuat menggunakan metode *Fuzzy Takagi-Sugeno* untuk melakukan proses klasifikasi terhadap kualitas air di kolam lele. Dengan menggunakan *microcontroller* Arduino Uno sebagai alat untuk memproses data dari dua sensor. *Output* dari proses sistem ini berupa aksi dari pompa yang disiapkan untuk melakukan menambah air ketika suhu dan kekeruhan air kolam melewati batas yang telah ditentukan. Selain menambahkan air ke kolam lele, pompa air bisa digunakan untuk menguras air ketika volume air berlebih. *Output* dari Arduino selain menjalankan pompa air, selain itu *output* dari sistem juga berupa mengirim data hasil proses *defuzzifikasi* ke *smartphone*. Proses mengirim data arduino ke *smartphone* menggunakan media *bluetooth* dengan modul HC-05. Waktu komputasi dari sistem ini dalam melakukan proses klasifikasi *fuzzy* dan mengirim data ke *smartphone* sekitar 5 hingga 10 detik. Kekurangan dari penelitian ini yaitu sistem yang dibuat menggunakan metode *fuzzy* untuk klasifikasi kualitas air kolam lele yang ketika menentukan hasil klasifikasinya memerlukan parameter di semua kondisi dari setiap *rule* harus terpenuhi.

### 2.1.3 Analisis Kualitas Air Dan Strategi Pengendalian Pencemaran Air Sungai Metro di Kota Kepanjen Kabupaten Malang

Pada penelitian ketiga yang dijadikan referensi dengan judul “Analisis Kualitas Air Dan Strategi Pengendalian Pencemaran Air Sungai Metro di Kota Kepanjen Kabupaten Malang” ditujukan untuk menganalisa status mutu air sungai metro dengan metode indeks pencemaran air. Pada penelitian ini metode untuk mendapatkan data tentang air sungai menggunakan pengambilan sampel air sungai di tiga titik. Ketiga titik tempat pengambilan sampel berada di desa Kedung Monggo, desa Talangagung dan desa Mangunrejo. Metode pengumpulan data dilakukan secara observasi lapangan, wawancara ke instansi yang berwenang dan dokumentasi. Parameter air yang diambil untuk penelitian berupa temperatur air, TTS(*Total Suspended Solid*), pH air, DO(*Disolved Oxygen*), COD(*Chemical Oxygen Demand*), PO<sub>4</sub>-P(*Phospat*), NO<sub>2</sub>-N(*Nitrit*), NO<sub>3</sub>-N(*Nitrat*), NH<sub>3</sub>-N(*Amonia*), BOD(*Biological Oxygen Demand*). Penelitian ini melakukan analisa sampel air di Laboratorium Perum Jasa Tirta 1 Malang dengan standar Laboratorium SNI. Kekurangan dari Penelitian ini yaitu proses menganalisa terhadap kualitas air sungai metro di Kepanjen masih dilakukan di Laboratorium, sehingga tidak semua masyarakat bisa menganalisa kualitas air. Belum adanya alat yang *portable* yang bisa digunakan oleh masyarakat biasa untuk menganalisa dan mengukur kualitas air sungai metro di Kepanjen. Pada penelitian ketiga ini dijadikan sebagai acuan dari parameter tingkat kualitas berdasarkan suhu, pH.

### 2.1.4 Real-Time Water Quality Monitoring System Using Internet of Things

Pada penelitian keempat yang dijadikan referensi berjudul “*Real-Time Water Quality Monitoring System Using Internet of Things*” berisi tentang penelitian *monitoring* kualitas air yang menggunakan parameter suhu air, pH air dan konduktivitas air. Pada penelitian ini menggunakan tabel WHO terhadap indeks kualitas air. LPC *microcontroller* digunakan sebagai untuk mengolah data dari sensor pada penelitian ini. Data yang didapat akan dikirim ke *smartphone* melalui sms



menggunakan modul GSM secara *real time*, Selain itu data yang didapat akan dikirim ke *cloud ThingSpeak* menggunakan modul node MCU. Pada *ThingSpeak* data yang diterima dapat divisualisasikan menjadi sebuah grafik. Dengan adanya grafik akan mempermudah si *user* dalam melakukan *monitoring* kualitas air secara *real time*. Kelebihan dari penelitian ini yaitu data yang didapat dari sensor akan diproses dan dikirim ke *smartphone* dengan modul GSM dan dikirim ke *cloud ThingSpeak* secara *real time*. Kekurangan dari penelitian ini belum adanya metode untuk melakukan klasifikasi kualitas air dengan parameter yang di dapat dari sensor.

### 2.1.5 Implementasi Algoritme *K-Nearest Neighbor* Untuk Identifikasi Kualitas Air (Studi Kasus: Pdam Kota Surakarta)

Pada penelitian kelima yang dijadikan referensi berjudul “Implementasi Algoritme *K-Nearest Neighbor* Untuk Identifikasi Kualitas Air (Studi Kasus: Pdam Kota Surakarta)” yang berisi tentang sebuah penelitian yang tentang kualitas air menggunakan metode k-NN. Pada penelitian ini menggunakan data yang didapatkan dari laboratorium PDAM kota Surakarta melalui hasil wawancara dan observasi langsung di tahun 2016. Parameter yang diteliti berupa Bau, Rasa, Suhu, Warna, Kekeruhan, Sisa Clor, Suhu, Ph, Fe, Mn,  $\text{NH}_4^+$ ,  $\text{NO}_2$ ,  $\text{KMnO}_4^-$ , CR,  $\text{SO}_4^{2-}$ . Dalam proses klasifikasi yang menggunakan metode k-NN mengambil 7 data terdekat ( $K = 7$ ) dalam proses menentukan hasil kualifikasinya. Hasil dari penelitian ini memiliki akurasi 82.5%. Pada penelitian ini memiliki kekurangan data yang diuji harus dilakukan di laboratorium dan masih secara manual. Penelitian ini memiliki kelebihan yaitu jumlah parameter yang sejumlah 15 parameter yang bisa mewakili kondisi air yang sebenarnya. Penelitian kelima ini akan dijadikan referensi dalam melakukan proses klasifikasi kualitas air dengan metode k-NN.

Dengan penelitian yang berjumlah lima yang berada di atas akan dijadikan referensi untuk penelitian ini. Pada penelitian ini akan membuat sistem yang bisa melakukan klasifikasi kualitas air bendungan dengan metode k-NN yang hasilnya bisa diakses menggunakan *smartphone* android. Sistem yang dibuat merupakan alat yang ditempatkan di bendungan yang mengukur kualitas air dengan parameter suhu air, pH, kekeruhan air, kedalaman air dan kandungan jumlah zat yang terlarut (*Total Disolved Solid*). Data hasil akuisisi sensor yang ada menggunakan *microcontroller* arduino uno dan *microcontroller* node MCU untuk melakukan klasifikasi dengan metode k-NN. Hasil dari klasifikasi dan hasil data akuisisi sensor akan dikirim ke firebase agar data hasil tersebut bisa diakses oleh pengguna melalui aplikasi *smartphone* android. Kelebihan dari sistem pada penelitian ini yaitu hasil dari klasifikasi air dapat diakses dengan aplikasi *smartphone* sehingga dapat memantau kondisi air di bendungan.

## 2.2 Dasar Teori

Dasar teori berisi tentang teori-teori yang berhubungan dengan perancangan penelitian dengan topik klasifikasi dan *monitoring* kualitas air dengan metode k-NN.



### 2.2.1 Karakteristik Air

Air memiliki beberapa karakteristik yang digunakan dalam uji coba dalam penelitian ini. Peraturan Pemerintah No.20 tahun 1990 mengelompokkan kualitas air menjadi beberapa golongan yaitu Golongan A, Golongan B, Golongan C, dan Golongan D. Selain itu dalam penelitian ini menggunakan air bendungan untuk tempat pengujiannya.

#### 1. Kekeruhan Air

Kekeruhan adalah jumlah butir-butir zat yang tidak bisa dilihat dengan mata telanjang yang tergenang dalam air, Kekeruhan biasanya terjadi karena adanya bahan organik dan anorganik yang tersuspensi dan terlarut (lumpur dan pasir halus)(Muthuraman G, 2014). Sedangkan dengan organik dan anorganik yang berupa mikro organisme dan plankton. Kekeruhan dalam air minum tidak boleh melebihi 5 NTU (*Nephelometric Turbidity Unit*) 4 (Permenkes RI, nomor 416/MENKES/PER/IX/1990, tentang syarat-syarat dan pengawasan kualitas air).

Kondisi di sekitar bendungan dan waduk memiliki tingkat kekeruhan yang berbeda beda, sehingga belum ada patokan untuk melakukan penelitian di tempat mana. Sensor yang digunakan memiliki cara kerjanya yaitu di dalam sensor terdapat bagian *sender* untuk mengeluarkan cahaya dan yang satunya *receiver* berfungsi untuk menerima cahayanya. Nilai kekeruhan air diukur dari cahaya yang diterima oleh salah satu bagian dari sensor. Ketika airnya jernih maka cahaya yang dipancarkan maka akan diterima oleh *receiver* dengan baik.

#### 2. pH Air

pH adalah kepanjangan dari *potensial hidrogen*. pH sendiri merupakan suatu ukuran konsentrasi dari ion hidrogen di sebuah larutan air. Dengan pengukuran pH akan bisa menentukan sebuah larutan bisa bersifat asam atau alkali (basa). Ph sendiri salah satu parameter faktor lingkungan yang bisa mempengaruhi pertumbuhan dan kehidupan di dalam air. Air murni bersifat netral dengan kadar pH 7. Air yang memiliki larutan pH kurang dari 7 bisa disebut air yang bersifat asam dan ketika air yang memiliki larutan pH lebih dari 7 bisa disebut air yang bersifat basa atau alkali. Kondisi air yang baik menurut peraturan pemerintahan RI no 82 tahun 2001 memiliki rentang nilai 6 hingga 9.

Terdapat beberapa aktivitas manusia dalam kegiatan sehari hari yang dapat membuat air memiliki tingkat pH yang sangat tinggi. Contoh Kegiatannya seperti kegiatan mencuci baju, kegiatan mandi. Dari beberapa contoh kegiatan di atas dapat mengakibatkan kualitas air dengan pH tinggi karena limbah air bekas kegiatan mengandung banyak senyawa kimia. Dampak dari air yang mengandung kadar pH yang tinggi akan mengakibatkan dampak yang buruk bagi kelangsungan hidup hewan yang berada di air.

#### 3. Suhu Air

Suhu di air adalah salah satu faktor yang penting bagi proses menentukan kualitas air di suatu daerah. Suhu sendiri sangat sering berubah dari tinggi ke rendah dengan berbagai penyebab seperti teriknya sinar matahari di daerah tertentu,



kedalaman dari air, kegiatan pembuangan air limbah dengan suhu tinggi dengan jumlah banyak juga mempengaruhi suhu di suatu perairan. Suhu air sangatlah mempengaruhi kegiatan biologi makhluk hidup yang berada di dalam air.

Dengan suhu yang naik di suatu daerah akan menyebabkan dampak *negative* seperti tumbuh dan berkembangnya beberapa jenis alga beracun contohnya kelompok *Cynophyta*. Selain itu dampak dari suhu yang meningkat akan membuat pengaruh secara langsung atau tidak langsung bagi hewan yang berada di dalam air seperti ikan, kepiting kecil (Mahida, 1993).

#### 4. Kedalaman Air

Kedalaman air adalah salah satu parameter fisika dalam menentukan kualitas air. Dengan kedalaman air akan bisa diketahui cahaya matahari yang masuk ke dalam air seberapa banyaknya karena banyak makhluk hidup seperti ikan sangatlah memerlukan cahaya matahari yang masuk ke dalam air. Cahaya matahari yang masuk ke air akan berhubungan dengan kandungan oksigen di dalam air tersebut dan suhu air tersebut. Sehingga kedalaman air sangatlah berpengaruh dengan parameter yang lainnya. Di dalam waduk ataupun bendungan akan memiliki tingkat kedalaman air yang berbeda di setiap tempat.

Pada penelitian ini menggunakan sensor sonar JSN-SR04T untuk melakukan pengukuran kedalaman air yang memiliki cara kerja sensor akan mengirim gelombang ultrasonik dan gelombang tersebut akan dipantulkan kembali oleh objek yang terdeteksi (Andang, 2019). Waktu dari ketika sensor memancarkan hingga menerima kembali gelombang ultrasonik tersebut akan dihitung untuk menentukan panjang objek yang terdeteksi.

#### 5. Jumlah Zat Yang Terkandung di dalam Air

Jumlah zat yang terlarut di dalam air biasanya disebut dengan TDS dengan nilai satuan ppm. Ppm (*Part Per Miliun*) merupakan satuan yang digunakan untuk mengukur konsentrasi larutan di air. Menurut PP NO.20 TAHUN 1990 jumlah TDS di dalam air yang baik adalah maksimum 500 ppm. Jumlah TDS di dalam air semakin kecil semakin baik, tetapi ketika air memiliki jumlah TDS yang mendekati 0 malah menjadi air yang tidak baik karena manusia memerlukan zat-zat tertentu yang ada di air ketika dia minum air tersebut. Salah satu kadar zat yang di dalam air yang diperlukan adalah kalsium (*Ca*), Magnesium (*Mg*) dan kalium. Jika manusia meminum air yang tidak ada mineral akan menyebabkan sering buang air kecil (nazava.com, 2019).

#### 2.2.2 Bendungan Air

Bendungan adalah sebuah cekungan yang berisi air yang terbuat secara alami ataupun secara buatan. Cekungan yang berisi air yang lebih kecil biasanya disebut dengan bendungan. Bendungan menurut Peraturan Pemerintah No 37 Tahun 2011 merupakan wadah dibuat dengan sengaja dengan tujuan untuk menampung air, menampung lumpur.



**Gambar 2.1 Gambar Bendungan Air**

Sumber : (mainichi.jp,2019)

Waduk dan bendungan memiliki fungsi yang sama yaitu sebagai saluran irigasi bagi para petani, sebagai pengendali banjir. Selain itu bendungan air seperti Gambar 2.1 memiliki fungsi sebagai tempat PLTA, sebagai penyedia air baku dan bisa sebagai wahana rekreasi dan pariwisata. Hanya saja ukuran atau luas daerah cekungan air yang membedakan waduk dan bendungan di mana bendungan memiliki luas wilayah lebih kecil ketimbang waduk.

### 2.2.3 Sensor Kedalaman Air

Sensor sonar JSN-SR04T yang digunakan pada penelitian ini yang berfungsi bisa mengukur kedalaman air untuk bendungan. Bendungan yang dangkal memungkinkan air yang mengalirnya akan bercampur dengan bahan organik dan anorganik sehingga membuat kejernihan airnya kurang. Sensor yang akan digunakan adalah sonar sensor JSN-SR04T.



**Gambar 2.2 Sensor JSN-SR04T**

Sumber : (forum.arduino.cc,2016)

Seperti pada Gambar 2.2 Sensor JSN-SR04T merupakan sensor ultrasonik yang memiliki cara kerja sensor akan mengirim gelombang ultrasonik dan gelombang tersebut akan dipantulkan kembali oleh objek yang terdeteksi (Andang, et al., 2019). Waktu dari ketika sensor memancarkan hingga menerima kembali gelombang ultrasonik tersebut akan dihitung untuk menentukan panjang objek yang terdeteksi.



#### 2.2.4 Sensor pH air

Sensor pH0-14 yang berfungsi untuk mengukur kadar pH di dalam air. Seperti pada Gambar 2.3 merupakan sensor pH0-14 yang dipakai pada penelitian ini. pH sendiri merupakan derajat keasaman yang digunakan untuk menyatakan tingkat keasaman dan basa yang dimiliki oleh suatu larutan. Air yang baik untuk konsumsi memiliki nilai pH 6,5 - 8,5 (Permenkes RI, nomor 907/MENKES/SK/VII/2002 tanggal 29 Juli 2002, tentang syarat - syarat dan pengawasan kualitas air minum).



Gambar 2.3 Sensor pH

Sumber : (wiki.dfrobot.com,2016)

#### 2.2.5 Sensor Suhu DS18B20

Sensor suhu DS18B20 yang digunakan di sistem pada penelitian ini dengan fungsi untuk mengukur suhu di dalam air. Suhu merupakan salah satu aspek dalam kualitas air. Karena pentingnya kondisi suhu air maka seperti pada Gambar 2.4 Sensor Suhu DS18B20 yang merupakan sensor digital yang memiliki 12-bit ADC internal. Sangat presisi, sebab jika tegangan referensi sebesar 5 Volt, maka akibat perubahan suhu, ia dapat merasakan perubahan terkecil sebesar  $5/(2^{12}-1) = 0.0012$  Volt. Pada rentang suhu  $-10^{\circ}\text{C}$  sampai  $+85^{\circ}\text{C}$ , sensor ini memiliki akurasi  $\pm 0.5$  derajat (github.com, 2019).



Gambar 2.4 Sensor Suhu DS18B20

Sumber : (kl801.ilearning.me,2017)



### 2.2.6 Sensor Kekeruhan Air

Sensor *Turbidity* digunakan di dalam sistem ini yang merupakan sensor yang memanfaatkan cahaya yang dipancarkan pada LED. Semakin tinggi tingkat kekeruhan air yang dideteksi maka tingkat pantulan cahaya yang diterima maka akan sedikit dan sebaliknya.



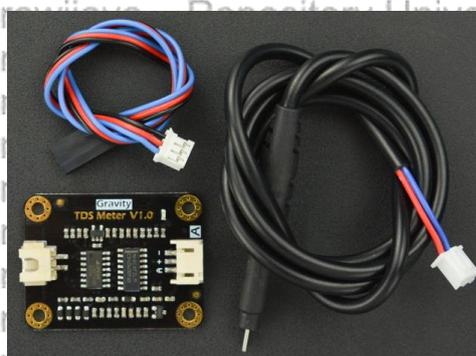
**Gambar 2.5 Sensor Turbidity**

Sumber : (teachmemicro.com, 2018)

Seperti pada Gambar 2.5 Sensor *Turbidity* di sini digunakan untuk mengukur kualitas air dari aspek kejernihan air (teachmemicro.com, 2018). Prinsip kerja dari sensor kekeruhan ini sama halnya dengan sensor proximity karena terdapat LED *photodiode* sebagai *transmitter* dan *photodiode (receiver)*.

### 2.2.7 Sensor TDS (Total Dissolved Solids)

Sensor TDS yang digunakan di penelitian ini untuk mengukur jumlah zat yang terlarut di dalam air. Sensor yang digunakan ber *type* TDS Meter V1.0 buatan dfrobot. Menurut PP NO.20 TAHUN 1990 jumlah TDS di dalam air yang baik adalah maksimum 500 ppm. Ppm (*Part Per Million*) merupakan satuan yang digunakan untuk mengukur konsentrasi larutan di air. Semakin banyak jumlah zat yang terlarut di dalam air mengindikasikan kualitas yang semakin jelek. Tetapi air dengan jumlah zat yang kurang dengan nilai dibawah 10 ppm juga tidak baik karena manusia memerlukan zat-zat yang terkandung di dalam air walaupun harus sesuai dengan kadar. Sensor TDS seperti pada Gambar 2.6 di atas mampu mengukur konsentrasi air dengan *range* 0 – 1000 ppm. Sensor TDS memiliki 3 pin yaitu pin *Vin* 5V, pin *ground* dan pin *output* data analog.



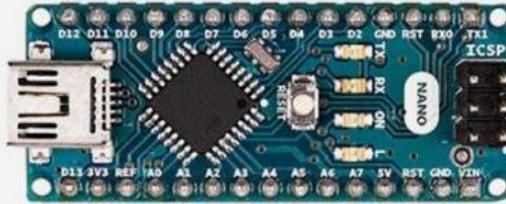
**Gambar 2.6 Sensor TDS**

Sumber : (df.robot.com,2019)



### 2.2.8 Arduino Nano

Arduino Nano adalah salah satu *product* yang dibuat oleh Arduino. Seperti pada Gambar 2.7 Arduino Nano adalah *board microcontroller* berbasis ATmega328 (*datasheet*). Memiliki 14 pin *input* dari *output* digital di mana 6 pin *input* yang dapat digunakan sebagai *output PWM* dan 8 pin *input* analog. Arduino nano memiliki ukuran 18 x 45mm yang sangatlah kecil tetapi tidak mengurangi fungsionalnya seperti arduino lainnya. Arduino Nano menggunakan USB mini-B untuk mendukung *microcontroller* agar dapat digunakan (arduino.cc, 2019).



Gambar 2.7 Arduino Nano

Sumber : (store.arduino.cc,2019)

### 2.2.9 Node MCU

Node MCU merupakan salah satu *microcontroller* yang banyak digunakan dalam berbagai penelitian yang berhubungan dengan IoT. Hal tersebut karena pada node MCU memiliki modul WiFi esp8266 yang berfungsi untuk mengirim data menggunakan koneksi internet. Node MCU seperti Gambar 2.8 memiliki kapasitas flash *memory* yang cukup besar yaitu 4 Mbytes dan *memory* SRAM sebesar 81 kBytes. Node MCU memiliki 16 pin *input output* dengan pin Analog 1 pin A0 (nodemcu.com, 2019).



Gambar 2.8 Node MCU

Sumber (nodemcu.com,2019)

### 2.2.10 Firebase

Firebase yang merupakan salah satu layanan dari Google yang fungsinya digunakan untuk mengembangkan aplikasi dalam menyimpan data bagi para pengembang. Berikut gambar dari logo Firebase seperti pada Gambar 2.9 Layanan yang ada di firebase seperti *Firestore* dan layanan *Real time*



Database. Selain dua layanan di atas ada salah satu layanan pendukung yaitu *Firestore* dan *Firebase Notification*.



**Gambar 2.9 Firebase**

Sumber: (firebase.google.com,2019)

Layanan *firebase Real time Database* memberikan fitur kepada pengembang aplikasi untuk memungkinkan data di aplikasi dilakukan sinkronisasi di seluruh klien dan data dapat disimpan di *firebase cloud*. *firebase* menyediakan layanan yang memungkinkan integrasi dengan berbagai platform seperti *Android, iOS, JavaScript, Java, Swift* dan *Node.js* aplikasi. *firebase Real time Database* merupakan *database* yang di *host* pada *cloud*. Data disimpan sebagai *JSON* dan disinkronkan secara *real time* ke setiap klien yang terhubung (firebase, 2019).

### 2.2.11 Arduino IDE

*Arduino IDE* yang merupakan perangkat lunak sumber terbuka untuk mempermudah menulis kode dengan bahasa pemrograman *C/C++* dan melakukan *upload* data ke board *arduino*. Berikut gambar dari *Arduino IDE* seperti pada Gambar 2.10. *Arduino IDE* berjalan pada sistem operasi *Windows, MAC OS X*, dan *Linux*. Perangkat lunak ini dapat digunakan dengan *board* *arduino* apa pun (*Arduino, 2019*). Konfigurasi dan penggunaan perangkat lunak *Arduino* sangat mudah dilakukan. Pada *arduino IDE* telah dilengkapi sejumlah pustaka (*library*) pendukung sehingga pengguna *Arduino* tidak harus membuat *library* yang baru. Pada *Arduino IDE* juga telah dilengkapi dengan *tutorial* dan sejumlah contoh yang dapat langsung digunakan (Rangkuti S, 2016).



**Gambar 2.10 Tampilan awal pada Arduino IDE**

Sumber: (www.labelektronika.com, 2017)



### 2.2.12 Android Studio

Android Studio yang merupakan sebuah IDE untuk pemrograman android secara resmi yang berasal dari google dengan pihak pengembang IntelliJ. Android studio didesain dan dirancang untuk pengembangan secara khusus android. Berikut logo dari Android Studio seperti pada Gambar 2.11. Android studio merupakan salah satu IDE yang bisa digunakan di berbagai OS seperti Windows, Linux dan Mac Os X. Android studio sangatlah cocok untuk para pengembang android level dasar karena banyaknya fitur yang sangatlah membantu.



Gambar 2.11 Android Studio

Sumber : (www.codepolitan.com, 2017)

Salah satu fitur dari android studio adalah fitur *intelligent code editor*, fitur ini memberikan kemudahan dalam analisis kode dan selain itu android studio menyediakan saran kode yang akan digunakan dengan sistem auto *complete*. ketika user sedang mengetik sebuah kode, aplikasi Android Studio mempunyai fitur untuk memberikan saran secara otomatis jika kode sudah dijalankan. Dengan beberapa fitur yang ada membuat si pengguna android studio dapat mempercepat waktu pembuatan program (codepolitan, 2017).

### 2.2.13 Metode *k-Nearest Neighbor*

Metode *k-Nearest Neighbor* merupakan metode yang termasuk *supervised learning* yang berarti metode yang perlu adanya pelatihan terlebih dahulu. Nilai akurasi klasifikasi dari metode k-NN sangat berpengaruh dari pemilihan nilai K yang menjadi penentu jumlah tetangga paling dekat (Wu, 2009).

#### 1. Proses Perhitungan Metode k-NN sebagai berikut :

1. Menentukan nilai K sebagai parameter jumlah tetangga paling dekat.
2. Menghitung kuadrat jarak data *testing* dengan data latih dengan rumus *Euclidean* dengan persamaan (2.1) sebagai berikut :

$$D = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_n - y_n)^2} \quad (2.1)$$

Keterangan :

D adalah Hasil jarak data latih dengan data uji yang didapatkan oleh kelima sensor.

$x_n$  adalah Data Uji ke-n yang didapat dari kelima sensor yang digunakan.



$y_n$  adalah Data Latih ke- $n$  yang sudah disiapkan dari kelima sensor yang digunakan.

3. Melakukan *sorting* data hasil perhitungan dari terkecil hingga terbesar.
4. Mengambil nilai terkecil sejumlah  $K$  dan selanjutnya menjadikan kelas nilai tertinggi di kelasnya dijadikan sebagai label kelas untuk data baru tersebut.

## 2. Proses Perhitungan *Weighted Voting* di metode K-NN :

1. Melakukan pengambilan  $K$  data terkecil / terdekat lalu di hitung dengan cara saling menjumlah data yang memiliki golongan yang sama.
2. Melakukan proses perhitungan *weighted voting* dengan persamaan 2.2 :

$$\text{Vote (golongan)} = \frac{1}{\text{data } x^2} + \frac{1}{\text{data } x^2} \approx \text{Hasil} \quad (2.2)$$

3. Hasil yang didapat akan dibandingkan dan nilai yang terbesar akan diambil dan dijadikan *output* klasifikasi



## BAB 3 METODOLOGI

Pada bab metodologi berisi penjelasan tentang metode yang akan digunakan. Pada bab ini juga terdapat isi yang menjabarkan tujuan dan cara dari setiap langkah langkah yang akan dilakukan dalam penelitian.

### 3.1 Tipe Penelitian

Penelitian dengan judul “Purwarupa Perahu untuk *Monitoring* dan Klasifikasi Kualitas Air Bendungan dengan Metode *K-Nearest Neighbor* (KNN)” merupakan penelitian dengan tipe penelitian implementatif pengembangan. Pada penelitian ini dilakukan pengembangan dengan jumlah sensor yang digunakan dengan jumlah 5, menggunakan metode lain dan objek yang diteliti yaitu bendungan air. Sensor yang digunakan pada penelitian ini yaitu sensor TDS, sensor suhu, sensor pH, sensor kedalaman dan sensor kekeruhan air. Dengan jumlah parameter kualitas air tersebut diharapkan dapat menggambarkan kualitas air yang sebenarnya .

### 3.2 Strategi Dan Rancangan Penelitian

Pada sub bab strategi dan rancangan penelitian ini berisi lokasi penelitian, teknik dalam pengumpulan data dan peralatan pendukung yang digunakan dengan tujuan agar penelitian ini menjadi lebih terstruktur dan lebih fokus.

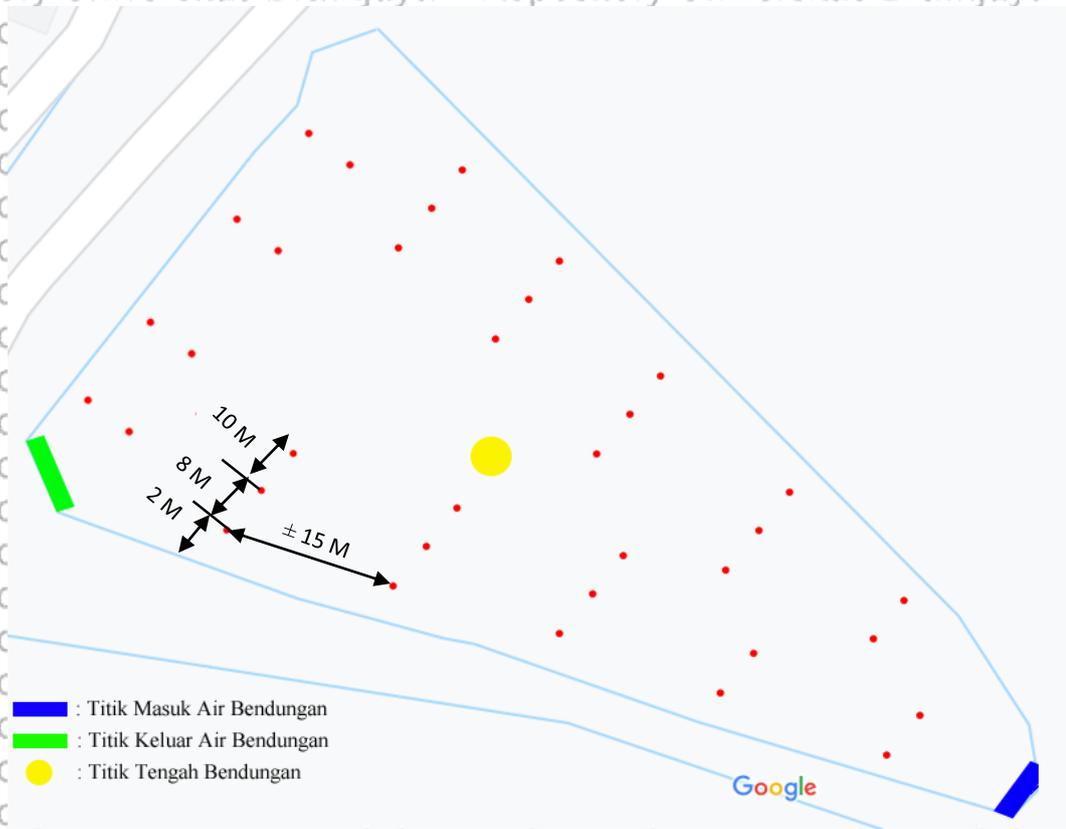
#### 3.2.1 Lokasi Penelitian

Lokasi penelitian yang digunakan memiliki 3 tempat yaitu bendungan air di desa Wungu Kab Madiun, bendungan air di PLTA desa Golang Kab Madiun dan bendungan air di PLTA desa Giringan Kab Madiun. Pemilihan ketiga tempat yang dilakukan penelitian ini dikarenakan fungsional dari bendungan air untuk masyarakat daerah sekitar sangat penting untuk berbagai kegiatan seperti kegiatan rumah tangga, kegiatan pertanian/perkebunan, untuk perikanan dan untuk peternakan. Selama ini proses *monitoring* kualitas air pada bendungan air tersebut kurang diperhatikan karena masih belum optimalnya fungsi dari bendungan tersebut. Sehingga lokasi tersebut dinilai tepat oleh peneliti untuk melakukan kegiatan pengujian dari penelitian ini agar diketahui kualitas air bendungan di ke 3 tempat tersebut. Ketika penelitian ini dapat dikerjakan dan dapat berfungsi sesuai dengan yang diinginkan peneliti maka alat ini dapat digunakan untuk warga sekitar dalam melakukan *monitoring* kualitas air dari 3 bendungan air tersebut.

#### 3.2.2 Teknik Pengumpulan Data

Penelitian ini dalam teknik pengumpulan data menggunakan cara melakukan sampling data dari kelima sensor terhadap air di bendungan dengan jarak antar titik yang sudah ditentukan sebelumnya. Selain melakukan sampling data maka dilakukan pengambilan sampel data air dari titik-titik yang telah ditentukan lalu dilakukan validasi kualitas air tersebut oleh peneliti sendiri dan akan digunakan menjadi data latih sistem yang dibuat dalam proses klasifikasi k-NN. Titik-titik yang telah ditentukan adalah 2 M, 8M, 10M dari pinggir bendungan serta tidak lupa

untuk mengambil data pada titik masuk air, titik tengah bendungan dan titik keluar air bendungan. Titik-titik pengumpulan data dapat dilihat seperti pada Gambar 3.1 dibawah ini.



Gambar 3.1 Titik-titik Pengumpulan Data

### 3.2.3 Peralatan Pendukung Yang Digunakan

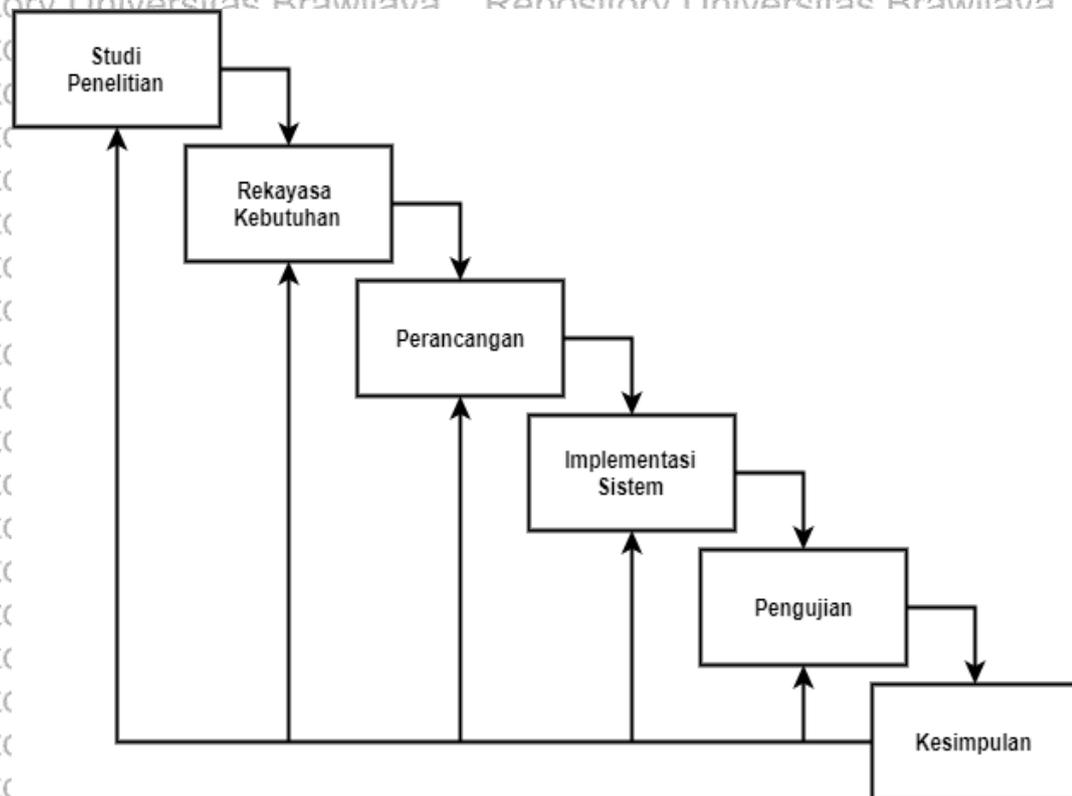
Peralatan pendukung digunakan untuk membantu dalam proses pembuatan sistem dan proses penelitian adalah node MCU, sensor TDS, arduino nano, sensor kedalaman, sensor pH, sensor kekeruhan dan sensor suhu. Node MCU digunakan untuk melakukan klasifikasi k-NN dengan data dari arduino nano dan untuk melakukan pengiriman data ke firebase agar data tersebut bisa ditampilkan aplikasi android. Arduino nano digunakan untuk melakukan akuisisi data dari kelima sensor yang digunakan dan mengirimkan data tersebut menuju node MCU agar bisa dilakukan proses selanjutnya. Selain itu, pada penelitian ini menggunakan laptop yang digunakan dalam proses pembuatan kode program untuk sistem yang dibuat dan untuk mengimplementasikannya.

## 3.3 Metode atau Teknik Lain

### 3.3.1 Isi Metodologi

Penelitian diawali dengan menentukan metode apa yang digunakan yang akan berguna untuk pelaksanaan penelitian. Metode yang akan digunakan dapat dipilih sesuai tujuan dari penelitian ini. Selain itu, beberapa hal yang akan dibahas di

metodologi seperti studi Penelitian, rekayasa kebutuhan, perancangan sistem, implementasi sistem, pengujian dan hasil analisis dan kesimpulan. Diagram alir yang digunakan menggunakan model *waterfall* yang bertujuan untuk mempermudah dalam proses pengerjaan skripsi ini jika dirasa hasil pengujian masih kurang. Tipe dari penelitian ini bertipe *design* implementatif pengembangan. Implementatif pada penelitian ini adalah dengan membangun sebuah alat untuk *monitoring* dan klasifikasi kualitas air bendungan dengan lima parameter berupa pH, suhu, kekeruhan, kedalaman, dan TDS (*Total Dissolved Solids*) dengan menggunakan metode k-NN dalam klasifikasi nya. Pengembangan pada penelitian ini adalah terdapat penelitian yang terdahulu adalah penggunaan sensor yang berjumlah 4, menggunakan metode *fuzzy* dengan objek sungai sehingga pada penelitian ini mengembangkan dengan menambah 5 sensor untuk memperkuat sistem dalam melakukan klasifikasi kualitas air, mengganti metode dengan metode klasifikasi k-NN untuk mengetahui akurasi sistem dengan metode klasifikasi k-NN dan melakukan penelitian dengan objek bendungan air. Langkah-langkah penelitian yang akan dilakukan seperti gambar diagram alir pada Gambar 3.2.



Gambar 3.2 Diagram Alir Pengerjaan Penelitian

### 3.3.2 Studi Literatur

Studi literatur digunakan untuk memahami dan mempelajari dasar teori yang akan digunakan. Dengan studi literatur bermanfaat untuk membantu agar tidak terjadi kendala saat perencanaan. Di dalam studi literatur, teori teori untuk mempelajari dan memahami bisa berasal dari buku, jurnal, dokumentasi, *e-book*



dan penelitian yang sebelumnya sudah dilakukan. Studi dalam penelitian ini yang akan dipakai adalah

1. Teori tentang klasifikasi k-NN.

Melakukan kegiatan studi literatur yang membahas proses dari metode klasifikasi k-NN.

2. *Monitoring* kualitas air

Studi literatur mengenai parameter yang berhubungan tentang air dalam proses menentukan kualitas air.

3. *Embedded* sistem

Mempelajari literatur tentang *embedded* sistem yang diimplementasikan ke dalam sebuah perangkat cerdas.

4. Arduino nano

Dilakukan studi literatur mengenai cara kerja, spesifikasi dan fungsional dari arduino nano.

5. Node MCU

Melakukan studi literatur tentang fungsional, cara kerja dan spesifikasi dari node MCU.

6. Sensor pH-014

Melakukan kegiatan studi literatur tentang spesifikasi sensor pH dan proses akuisisi data sensor yang digunakan.

7. Sensor suhu DS18B20

Bagian ini akan melakukan kegiatan studi literatur tentang cara akuisisi data dari sensor suhu yang digunakan beserta spesifikasinya.

8. Sensor kekeruhan air

Pada bagian ini mengetahui fungsional sensor yang digunakan beserta proses akuisisi data sensor kekeruhan dan spesifikasinya.

9. Sensor kedalaman JSN-SR04T

Mengetahui sensor kedalaman untuk memahami proses akuisisi data beserta spesifikasi sensor yang digunakan.

10. Sensor TDS

Melakukan kegiatan studi literatur untuk memakai sensor TDS dapat melakukan akuisisi data sensor beserta spesifikasinya.

11. Android Studio

Melakukan proses studi literatur dalam proses membuat aplikasi android beserta fungsi fungsi yang ada di dalam aplikasi android studio.



## 12. Firebase

Dilakukan proses memahami firebase dalam proses mengirim dan menerima data.

### 3.3.3 Rekayasa Kebutuhan

Rekayasa kebutuhan ditentukan dengan tujuan untuk menentukan dan mendapatkan semua kebutuhan yang diperlukan untuk merancang sistem yang akan dibangun. Rekayasa kebutuhan dapat dibagi ke beberapa kategori yaitu perangkat sistem yang akan di rancang, kebutuhan perangkat keras, kebutuhan perangkat lunak dan kebutuhan fungsional dan kebutuhan non fungsional sistem.

#### 1. Kebutuhan fungsional

1. Sistem dapat melakukan akuisisi data sensor pH-014
2. Sistem dapat melakukan akuisisi data sensor suhu air DS18B20
3. Sistem dapat melakukan akuisisi data sensor kedalaman air JSN-SR04T
4. Sistem dapat melakukan akuisisi data sensor kekeruhan air *Turbidity*
5. Sistem dapat melakukan akuisisi data sensor TDS (*total dissolved solids*)
6. Sistem dapat melakukan proses klasifikasi dengan metode k-NN dan menentukan hasil *output* dari sistem ketika mendapatkan *input* dari lima sensor yang digunakan.
7. Sistem dapat mengirim data sensor dan data klasifikasi ke *database* firebase.
8. Aplikasi android dapat menampilkan data dari firebase.
9. Sistem dapat menggerakkan alat mengelilingi bendungan untuk kegiatan *monitoring* kualitas air.

#### 2. Kebutuhan non-fungsional

##### 1. Kebutuhan Perangkat Keras

1. Arduino Nano dipilih untuk *microcontroller* pusat pemrosesan dari sistem berupa mengambil data sensor dan mengirim data sensor ke node MCU.
2. Node MCU bertugas untuk mengklasifikasikan data dengan metode k-NN dan mengirim data tersebut ke *database* melalui koneksi internet.
3. 1 baterai *Lippo* 12V digunakan untuk sumber daya untuk *microcontroller* arduino nano, node MCU dan kelima sensor yang digunakan.
4. Sensor Suhu DS18B20 dapat untuk melakukan akuisisi data suhu air.
5. Sonar Sensor JSN-SR04T digunakan untuk mengakuisisi data kedalaman air.
6. Sensor pH Arduino digunakan untuk mengakuisisi kondisi pH air.
7. Sensor *Turbidity* Arduino digunakan untuk mengakuisisi data kualitas air dari segi kekeruhannya.



8. Sensor TDS (*Total Dissolved Solid*) digunakan untuk mengakuisisi data kualitas air dari segi jumlah zat yang terlarut dalam air tersebut.

9. *Smartphone* android digunakan untuk melihat hasil pengukuran alat melalui aplikasi yang sudah dibuat.

## 2. Kebutuhan perangkat Lunak

1. Arduino IDE digunakan untuk tempat merancang program c untuk arduino.

2. Android Studio untuk perancangan dan implementasi program aplikasi berbasis android.

3. Eagle pada penelitian ini untuk merancang pcb dari sistem yang dibuat.

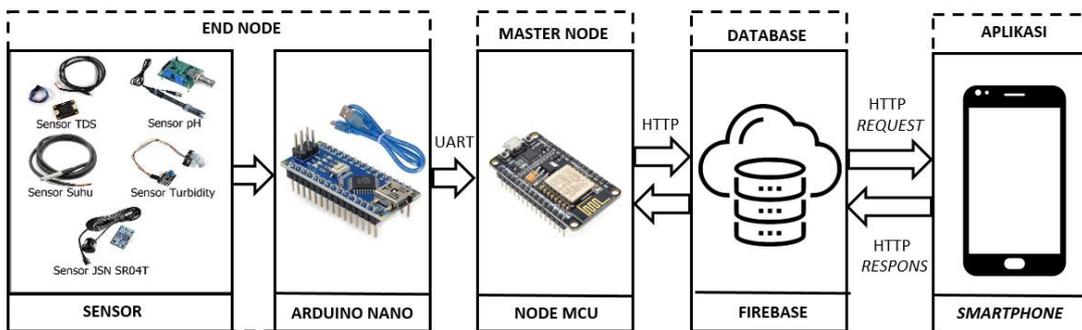
4. Browser Chrome untuk pembuatan dan pengecekan data di *database* firebase.

### 3.3.4 Perancangan

Pada bagian perancangan, pada penelitian ini akan melakukan perancangan sebuah sistem dengan 2 tahap yang terdiri dari perancangan sistem perangkat keras dan sistem perancangan perangkat lunak.

#### 1. Perancangan Sistem Perangkat Keras

Pada tahap perancangan sistem perangkat keras memiliki tujuan agar kinerja dari sistem dapat sesuai dengan harapan ketika proses perancangan. Perangkat keras pada penelitian ini menggunakan arduino nano dan node MCU sebagai *microcontroller*. *Input* data yang digunakan sistem adalah data dari 5 sensor seperti sensor kekeruhan, sensor TDS, sensor kedalaman air, sensor suhu dan sensor pH. Hasil dari sistem ini berupa data yang ditampilkan di dalam *smartphone* berbasis android. Penggunaan *smartphone* di sini yang harus terhubung dengan internet agar dapat mengakses data tersebut.



**Gambar 3.3** Blok Diagram Perancangan Sistem Perangkat Keras

Pada Gambar 3.3 blok diagram perancangan sistem perangkat keras dapat dijelaskan berupa poin poin sebagai berikut:

1. *End Node* adalah gabungan dari beberapa komponen seperti dari sensor pH, sensor *Turbidity*, sensor TDS (*Total Dissolved Solid*), sensor JSN-SR04T dan



sensor DS18B21 akan mengambil nilai di suatu bendungan. Ketika semua sensor sudah mendapatkan nilai di suatu bendungan, data akan diproses oleh *microcontroller* Arduino Nano untuk dikirim menuju node MCU.

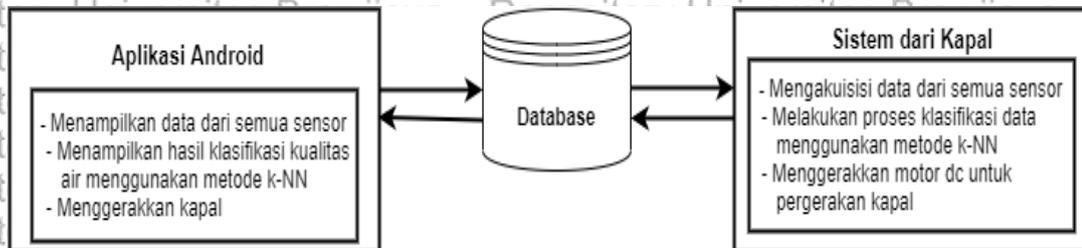
2. *Master Node* adalah perangkat yang berfungsi untuk menghubungkan *end node* dengan server. *Master node* di sini menggunakan node MCU yang bertugas untuk menerima data sensor dari arduino nano dan melakukan proses klasifikasi k-NN dan setelah itu mengirimkan ke *database* yaitu firebase.

3. Database yang digunakan pada penelitian ini adalah *Firestore* yang menggunakan *protocol* HTTP. *Firestore* digunakan untuk menyimpan data sensor dan hasil klasifikasi.

4. Aplikasi *Smartphone* di sini berfungsi untuk melakukan *request* data dari *firebase* yang ditampilkan di aplikasi berbasis android agar memudahkan pengguna untuk mengetahui data tersebut.

## 2. Perancangan Sistem Perangkat Lunak

Ketika perancangan perangkat keras sudah selesai dilakukan maka akan dilanjutkan ke tahap perancangan perangkat lunak. Perancangan perangkat lunak sangatlah penting karena dengan perancangan perangkat lunak, data yang didapat dari sensor dapat diolah dan diproses sehingga data hasil pemrosesan perangkat lunak datanya bersifat valid.



**Gambar 3.4 Blok Sistem Perancangan Perangkat Lunak**

Berikut merupakan tahapan perancangan perangkat lunak seperti Gambar 3.4:

1. Sensor yang dipakai akan mengambil data yang akan digunakan untuk data *training* dan data testing.
2. Data dari sensor pada awalnya akan dimasukkan ke dalam data *training* dengan label tiap kelasnya
3. Data *training* yang sudah didapatkan dan bersifat valid akan menjadi patokan untuk melakukan proses klasifikasi selanjutnya.
4. Sensor mengambil data kualitas air untuk dijadikan data testing. Data yang dihitung kedekatannya dengan data *training* menggunakan rumus *euclidean*.
5. Hasil dari proses *euclidean* akan dilakukan *ranking* dari data yang paling mendekati dan dihitung k-NN *weighted*.
6. Hasil dari proses k-NN *weighted* akan dikirim ke *database* agar data dapat tersimpan di *database*.



7. Data yang sudah tersimpan di *database*, dapat diakses melalui aplikasi di *smartphone* berbasis android.

8. Data yang diakses oleh aplikasi *smartphone* meliputi data hasil klasifikasi kualitas air, data dari tiap-tiap sensor dan menggerakkan alat.

### 3.4 Implementasi

Implementasi sistem dilakukan sesuai dengan perancangan sistem yang telah dibuat. Pada implementasi terdapat beberapa langkah proses pengerjaan seperti implementasi perangkat keras, implementasi perangkat lunak dan implementasi yang menghubungkan antara perangkat keras dan lunak.

Pada implementasi perangkat keras dimulai dengan melakukan pembuatan wadah / tempat untuk konfigurasi perangkat keras yang diperlukan. Penggunaan perangkat keras dalam penelitian ini meliputi arduino nano, node MCU, sensor ph air, sensor kekeruhan air, sensor kedalaman air dan sensor jumlah zat yang terlarut di dalam air. Untuk proses implementasi perangkat lunak dimulai dengan konfigurasi dari tiap perangkat keras menggunakan bahasa C. Setelah semua dilakukan maka masuk ke tahap implementasi penggabungan antara perangkat keras dengan perangkat lunak sebagai kendalinya.

### 3.5 Pengujian dan Analisis

Tahapan pengujian dan analisis digunakan untuk mengetahui performa dari alat yang dibuat. Performa yang bisa meliputi akurasi dalam klasifikasi dan waktu yang diperlukan untuk melakukan proses klasifikasi. Selain itu performa sistem berupa penggunaan *memory* pada *microcontroller*. Tahapan tahapan dari pengujian sistem ini adalah sebagai berikut:

1. Proses menguji akurasi dari sistem pada setiap sensor dalam mendapatkan nilai secara satu persatu.
2. Proses menguji sistem dalam mengklasifikasikan data yang didapat oleh tiap sensor dalam menentukan klasifikasi kualitas air.
3. Proses menguji waktu komputasi sistem dan penggunaan *memory* dari sistem yang dibuat menggunakan metode klasifikasi k-NN.
4. Proses menguji data pada aplikasi di *smartphone* dengan data yang dikirim oleh *microcontroller* untuk mengetahui apakah terdapat perbedaan antara data yang dikirim dari *microcontroller* hingga diterima oleh aplikasi android.
5. Proses menguji tingkat *reliabilitas* proses mengirim data ke *database* firebase dan menampilkan data pada aplikasi *smartphone* dari sistem yang dirancang.

Dalam penelitian ini sangat perlu untuk mengukur performa dari sistem yang dibuat karena agar dapat diketahui layak atau tidak untuk masyarakat secara langsung.



### 3.6 Kesimpulan

Proses pengambilan kesimpulan dilakukan ketika semua proses yang di atas telah selesai dilakukan. Prosesnya meliputi tahapan perancangan, implementasi dan pengujian sistem yang semuanya perlu dilakukan terlebih dahulu. Kesimpulan dapat diambil dari hasil pengujian yang telah dilakukan dan telah dianalisa terlebih dahulu. Tahap terakhir dari penulisan ini adalah memberikan saran yang bertujuan untuk memperbaiki atau menyempurnakan dari sistem yang sekarang agar ke depannya bisa lebih baik jika penelitian ini akan dikembangkan lagi dan untuk menjadi referensi penelitian lain. Saran juga dapat menjadi referensi untuk penelitian lain yang berhubungan dengan penelitian ini.



## BAB 4-REKAYASA KEBUTUHAN

### 4.1 Gambaran Umum Sistem

Pada penelitian ini akan membuat alat yang bisa melakukan *monitoring* dan klasifikasi kualitas air di bendungan secara otomatis. Dengan cara menempatkan alat yang bergerak sendiri dengan motor DC sesuai perintah dari *smartphone* untuk *monitoring* kualitas air. Alat yang dibuat ini menggunakan sensor TDS, sensor pH, sensor suhu air, sensor kedalaman air dan sensor kedalaman air yang digunakan untuk sebagai parameter menentukan kualitas air. Kelima sensor tersebut akan sebagai *input* untuk *microcontroller* arduino nano sebagai pemroses utama. Data dari akuisisi kelima sensor yang diterima oleh arduino nano akan dikirim ke node MCU. Alat yang dibuat di penelitian ini menggunakan node MCU yang berfungsi untuk melakukan klasifikasi kualitas air menggunakan metode k-NN sehingga dapat diketahui kualitas air dari hasil klasifikasi tersebut dan berfungsi untuk mengirimkan data hasil klasifikasi yang telah diproses dan data akuisisi sensor ke *database firebase*. Data yang dikirim ke *firebase* akan diakses dan ditampilkan oleh aplikasi android yang telah dibuat agar data hasil kualitas air tersebut bisa langsung diterima dan dibaca oleh pengguna melalui *smartphone* mereka. Data yang bisa diakses dengan *smartphone* secara langsung bisa digunakan untuk bahan evaluasi tentang kualitas air yang di *monitoring* dan bisa sebagai informasi kualitas air bendungan di sekitar mereka.

### 4.2 Rekayasa Kebutuhan Sistem

Pada rekayasa kebutuhan sistem dilakukan untuk menjelaskan lebih spesifik dari kebutuhan yang diperlukan dan sistem yang digunakan akan bisa bekerja sesuai dengan keinginan dan tujuan dibuat alat tersebut. Rekayasa kebutuhan berisi tentang kebutuhan fungsional sistem dan kebutuhan non fungsional sistem. Kebutuhan non fungsional sistem di dalamnya memiliki uraian penjelasan kebutuhan perangkat lunak dan kebutuhan perangkat keras.

#### 4.2.1 Kebutuhan Fungsional

Kebutuhan fungsional berisi penjelasan tentang sebuah kebutuhan yang harus disediakan dan berisi proses-proses atau fungsi-fungsi yang harus dijalankan jika terdapat sebuah masukan tertentu. Di sistem yang dibuat ini menggunakan arduino nano sebagai pusat pengolah data. Berikut ini kebutuhan fungsional dari sistem seperti pada Tabel 4.1.

Tabel 4.1 Kebutuhan Fungsional Sistem

No	Kode	Kebutuhan Fungsional Sistem	Penjelasan
1	TS-01	Sistem dapat melakukan akuisisi data sensor pH air	Sensor yang digunakan pada penelitian ini salah satunya sensor pH yang memiliki <i>range</i> pH 0 - 14 .



No	Kode	Kebutuhan Fungsional Sistem	Penjelasan
			Sensor pH memberikan <i>output voltage</i> pada <i>microcontroller</i> ketika proses akuisisi data sehingga dilakukan proses perhitungan regresi linier nilai <i>voltage output</i> pada tiap pH yang dideteksi. Pada nilai hasil regresi linier yang didapat maka akan bisa diproses dengan metode klasifikasi k-NN agar diketahui kualitas air yang diuji.
2	TS-02	Sistem dapat melakukan akuisisi data sensor suhu air DS18B20.	Sensor DS18B20 merupakan sensor suhu yang digunakan untuk mengukur suhu air yang menjadi salah satu parameter fisika kualitas air. Nilai sensor tersebut memiliki <i>range</i> -10°C sampai +85°C. Nilai dari sensor DS18B20 akan diproses oleh <i>microcontroller</i> sebagai salah satu parameter klasifikasi kualitas air.
3	TS-03	Sistem dapat melakukan akuisisi data sensor kedalaman air JSN-SR04T	Sensor JSN-SR04T digunakan untuk mengukur kedalaman air dengan <i>range</i> sensor 20 CM hingga 450 CM. Nilai sensor JSN-SR04T akan diproses oleh <i>microcontroller</i> dan nilai tersebut sangatlah penting digunakan sebagai salah satu parameter kualitas air yang sedang di uji.
4	TS-04	Sistem dapat melakukan akuisisi data sensor kekeruhan air <i>Turbidity</i>	Sensor yang digunakan untuk mengukur kekeruhan air yaitu menggunakan sensor <i>turbidity</i> . Sensor <i>Turbidity</i> juga mengeluarkan <i>output voltage</i> setiap dari hasil pengukuran kekeruhan air. Sehingga diperlukan regresi linier untuk menentukan parameter kekeruhan air yaitu NTU. Nilai dari sensor <i>turbidity</i> digunakan untuk sebagai salah satu parameter



No	Kode	Kebutuhan Fungsional Sistem	Penjelasan
5	TS-05	Sistem dapat melakukan akuisisi data sensor TDS	klasifikasi metode k-NN untuk menentukan kualitas air yang sedang diuji. Sensor TDS digunakan untuk mengukur kualitas air dengan mendeteksi jumlah zat yang terlarut di dalam air yang di uji. Parameter TDS sangatlah penting karena jumlah zat yang terlarut di dalam air sangat berpengaruh dengan kualitas air yang baik. Nilai dari sensor TDS memiliki range 0-1000 PPM dan nilai tersebut akan diproses oleh <i>microcontroller</i> sebagai salah satu parameter klasifikasi kualitas air dengan metode k-NN.
6	TS-06	Sistem dapat melakukan proses klasifikasi dengan metode k-NN menentukan hasil <i>output</i> dari sistem ketika mendapatkan <i>input</i> dari lima sensor digunakan.	Metode k-NN merupakan metode yang menentukan hasil klasifikasi dari data yang diterima dengan mencari data yang paling mirip dengan data <i>training</i> dengan cara membandingkan ke semua data yang <i>training</i> . Hasil dari klasifikasi bisa digunakan untuk berbagai hal seperti untuk penelitian selanjutnya. Proses klasifikasi dilakukan oleh node MCU.
7	TS-07	Sistem dapat mengirim data sensor dan data klasifikasi menuju <i>database</i> firebase	Pada alat ini menggunakan node MCU sebagai alat untuk melakukan klasifikasi k-NN dengan data dari 5 sensor dan mengirim data tersebut agar bisa diakses oleh pengguna menggunakan aplikasi <i>smartphone</i> . Sistem di sini diharuskan memiliki fungsionalitas untuk melakukan proses pengiriman data dari arduino nano menuju firebase sebagai



No	Kode	Kebutuhan Fungsional Sistem	Penjelasan
8	TS-08	Sistem aplikasi android dapat menampilkan data dari <i>database</i> firebase	<i>database</i> melalui media perantara node MCU. Data yang ada di firebase hasil klasifikasi haruslah bisa diakses menggunakan aplikasi pengguna berbasis android sehingga data tersebut bisa digunakan untuk hal yang lain seperti untuk analisa kualitas air ke depannya.

#### 4.2.2 Kebutuhan Non Fungsional

Kebutuhan non fungsional merupakan sebuah batasan fungsional dari sistem yang telah dibuat agar sistem dibuat lebih terfokus dengan fungsionalnya. Berikut uraian kebutuhan non fungsional sebagai berikut :

##### 1. Kebutuhan Perangkat Keras (Hardware)

Analisa perangkat keras yang dibutuhkan pada penelitian ini di tujuan agar alat yang dibuat untuk penelitian ini sesuai dengan yang direncanakan di awal pembuatan. Perangkat yang dibutuhkan di penelitian ini seperti pada Tabel 4.2. Penggunaan dua *microcontroller* yaitu arduino nano dan node MCU dikarenakan node MCU memiliki spesifikasi hanya satu pin analog dan pada penelitian ini menggunakan 3 pin analog untuk sensor pH, TDS dan kekeruhan. Kebutuhan pin analog yang tidak tercukupi oleh node MCU, sehingga memerlukan satu *micorcontroller* yang lebih difungsikan sebagai akuisisi data sensor. Node MCU difungsikan untuk melakukan klasifikasi dengan metode k-NN dan sebagai pengirim data dari arduino nano menuju firebase dan berfungsi untuk menggerakkan motor di alat sesuai perintah dari aplikasi pengguna. Dengan data pada sistem dikirim ke firebase bertujuan agar data tersebut akan di tampilkan di aplikasi android pengguna.

##### 1. Arduino Nano

Arduino Nano digunakan sebagai pusat dari sistem yang dibuat. Arduino nano mempunyai tugas untuk mengakuisisi data dari kelima sensor yang digunakan. Data tersebut akan dikirim oleh arduino nano menuju node MCU untuk dilakukan proses klasifikasi dan untuk dikirim ke firebase. Kebutuhan sistem memerlukan pin Analog sebanyak 3 dan karena node MCU kurang pin analog sehingga dipilihlah arduino nano sebagai *microcontroller*. Arduino nano memiliki spesifikasi yang mencukupi untuk melakukan akuisisi data dari kelima sensor yang digunakan. Berikut spesifikasi dari arduino nano seperti pada Tabel 4.2.



**Tabel 4.2 Tabel Spesifikasi Arduino Nano**

Sumber : (www.arduino.cc/usa/arduino-nano, 2019)

Fitur	Keterangan
Microcontroller	Atmega328
Analog Pin Input	8 (A0-A7)
Digital I/O Pin	14 pin Digital
Flash Memory	32 KB (2 KB untuk <i>bootloader</i> )
SRAM	2KB
Operation Voltage	5V

2. Node MCU

Node MCU merupakan salah satu *microcontroller* yang memiliki modul ESP8266 dan di dalamnya terdapat modul WiFi. Node MCU dipilih pada penelitian ini untuk melakukan proses klasifikasi k-NN dan melakukan proses mengirim data kelima sensor menuju firebase serta meminta data dari firebase untuk menggerakkan motor DC. Node MCU memerlukan koneksi WiFi untuk melakukan tugasnya sehingga pada penelitian ini memerlukan *tethering* Hp / WiFi di sekitar bendungan agar node MCU bisa mengirim data menuju firebase dan meminta data untuk menggerakkan motor DC. Node MCU memiliki spesifikasi yang cukup untuk mengerjakan tugas sebagai *microcontroller* pada penelitian ini. Spesifikasi node MCU seperti pada Tabel 4.3.

**Tabel 4.3 Spesifikasi Tabel Node MCU**

Sumber : (www.instructables.com, 2019)

Fitur	Keterangan
Microcontroller	ESP8266
Operation Voltage	3,3V–5V
WiFi	IEEE 802.11 b/g/n
Flash Memory	4 MB
SRAM	81 KB

3. Sensor TDS

Sensor TDS merupakan sensor yang berfungsi untuk mengukur kadar kualitas air dari segi jumlah zat yang terlarut. Satuan dari TDS yaitu PPM (*part per million*). Di dalam air terdapat banyak zat yang terlarut walaupun air tersebut air bersih sehingga perlu diukur jumlah zat yang terkandung di

dalam air. Standar maksimal kadar zat yang terlarut menurut pemerintah Indonesia adalah 500 PPM. Spesifikasi dari sensor tds seperti pada Tabel 4.4.

**Tabel 4.4 Spesifikasi Sensor TDS**

Sumber : (www.dfrobot.com, 2019)

Fitur	Keterangan
Operation Voltage	5V
Range Nilai Output Sensor	0 - 1000 PPM.
Akurasi Nilai Sensor	± 10%
Pin	3 (VCC, ground, Pin Analog Data)

#### 4. Sensor Kedalaman Air

Sensor kedalaman air yang digunakan *bertype* JSN SR04T. Sensor kedalaman ini memiliki cara kerja yang sama dengan sensor ultrasonik lainnya. *Range* sensor ini adalah 20 CM – 450 CM yang bisa digunakan untuk mengukur kedalaman dari bendungan air. Sensor yang digunakan bersifat *water proof* sehingga bisa digunakan untuk di dalam air tetapi memiliki hasil akurasi yang berbeda ketika bekerja di atas air. Kecepatan rambat ultrasonik di dalam air dan di atas air memiliki perbedaan yaitu ketika di dalam air sebesar 1498 m/s dan ketika di atas air sebesar 343 m/s. Kecepatan rambat di dalam air 4,4 kali lebih cepat dari pada di atas air. sensor JSN SR04T hanya memiliki satu buah *transducer* yang di dalamnya sudah termasuk *transmitter* dan *receiver* dari gelombang ultrasonik. Cara kerja sensor ini yaitu menghitung waktu dari pancaran gelombang ultrasonik dari *transmitter* hingga gelombang tersebut diterima kembali oleh *receiver*. Waktu ketika waktu antara gelombang dikirim dan diterima dalam bentuk *microseconds*. Spesifikasi dari sensor kedalaman air seperti pada Tabel 4.5.

**Tabel 4.5 Spesifikasi Sensor Kedalaman Air**

Sumber : (www.makerguides.com/jsn-sr04t-arduino, 2019)

Fitur	Keterangan
Operation Voltage	5 V
Range Akuisisi Data Sensor	20 CM – 450 CM
Pin	4 (VCC,ground,Trigger dan Echo)

#### 5. Sensor Kekeruhan Air

Kekeruhan air merupakan salah satu faktor penentu kualitas air. Dengan pentingnya variabel kekeruhan air maka penelitian ini menggunakan sensor kekeruhan air. Cara kerja dari sensor kekeruhan adalah sensor memiliki 1 *transducer* sebagai *transmitter* dan 1 *transducer* sebagai *receiver*. Semakin keruh air maka akan menjadi penghalang sinar yang dipancarkan



oleh *transmitter* hingga diterima oleh *receiver*. Sensor kekeruhan air memiliki *output* data berupa analog. Spesifikasi dari sensor kekeruhan air seperti pada Tabel 4.6.

**Tabel 4.6 Tabel Spesifikasi Sensor Kekeruhan Air**

Sumber : (wiki.dfrobot.com/Turbidity\_sensor\_SKU\_SEN0189, 2019)

Fitur	Keterangan
<i>Operation Voltage</i>	5 V
Pin	3 ( <i>VCC</i> , <i>ground</i> dan Analog pin)
<i>Range Suhu Operasi</i>	0 – 45 °C

#### 6. Sensor Suhu Air

Suhu air menjadi salah satu variabel untuk menentukan kualitas air pada penelitian ini. Sensor suhu air yang digunakan *bertype* DS18B20 dengan *output* data digital. Sensor DS18B20 bersifat *water proof* sehingga dapat digunakan untuk mengukur suhu air dan sensor ini memiliki resolusi ADC 9 bit hingga 12 bit sehingga memiliki kepresisian data sebesar  $5/(2^{12}-1) = 0.0012$  volt. Karena pada arduino nano memiliki resolusi nilai ADC 10 bit maka sensor DS18B20 memiliki kepresisian data sebesar  $5/(2^{10}-1) = 0.0008$  volt. Spesifikasi dari sensor suhu air dapat dilihat seperti pada Tabel 4.7.

**Tabel 4.7 Spesifikasi Sensor Suhu**

Sumber : (lastminuteengineers.com/ds18b20-arduino/, 2019)

Fitur	Keterangan
<i>Operation Voltage</i>	5 V
<i>Range Suhu Operasi</i>	-55 °C hingga + 85 °C
Pin	3 ( <i>VCC</i> , <i>ground</i> , Pin Digital Data)

#### 7. Sensor pH Air

Sensor pH terdapat 2 jenis yaitu untuk mendeteksi pH air dan pH tanah. Sensor untuk mendeteksi pH air yaitu pH-014. Sensor pH terdapat *glass electrode* yang di dalamnya terdapat kawat listrik dan cairan kalium klorida. Di dalam sensor pH terdapat cairan pH 7 pada kalium klorida dan ketika sensor pH ingin mengukur pH air di luar maka sensor pH akan membandingkan nilai pH di dalam *glass electrode* dan di luar sehingga mendapatkan perbedaan *voltage* yang diterima oleh kawat listrik di dalam. Perbedaan tersebut dihasilkan dari perbedaan ion hydrogen dan perbedaan *voltage* tersebut akan digunakan untuk menilai pH air di luar *glass electrode*. Spesifikasi dari sensor pH air seperti pada Tabel 4.8.

**Tabel 4.8 Spesifikasi Sensor pH Air**

Sumber : (www.dfrobot.com, 2019)

Fitur	Keterangan
Operation Voltage	5 V
Range nilai sensor	0 – 14
Pin	3 (VCC,ground, Pin Analog Data)

8. Motor DC 12V

Motor DC berfungsi untuk menggerakkan alat yang dirancang pada penelitian ini. Jumlah motor DC yang digunakan berjumlah 2 dengan fungsi yang berbeda. 1 motor DC untuk menggerakkan alat ketika berbelok kanan dan yang 1 motor DC untuk menggerakkan alat ketika bergerak kiri. Untuk maju atau mundur maka 2 motor sekaligus akan bergerak sesuai dengan perintah. Spesifikasi motor DC yang digunakan seperti pada Tabel 4.9.

**Tabel 4.9 Spesifikasi Motor DC 12V**

Sumber : (www.nidec-servo.com/en/digital/b\_dme/dme25, 2019)

Fitur	Keterangan
Operation Voltage Input	12V
Kecepatan Rotasi	6000 rpm
Ukuran	42 mm x 30 mm

9. Driver Motor L298N

Driver Motor L298N digunakan untuk mengatur pergerakan motor DC sebagai penggerak purwarupa alat. Selain itu pada driver motor terdapat output 5V yang digunakan untuk memberikan daya microcontroller arduino nano, node MCU dan kelima sensor yang digunakan. Spesifikasi dari driver motor dapat dilihat pada Tabel 4.10.

**Tabel 4.10 Spesifikasi Driver Motor L298N**

Sumber : (www.handsontec.com, 2019)

Fitur	Keterangan
IC Driver Motor	L298N
Operation Voltage Input	12 V
Output Voltage	5 V

**2. Kebutuhan Perangkat Lunak (Software)**

Analisa perangkat lunak yang dibutuhkan pada penelitian ini di tujuan agar sistem pada alat yang dibuat untuk penelitian ini sesuai dengan yang direncanakan



di awal pembuatan. Perangkat lunak yang dibutuhkan di penelitian ini seperti di bawah ini.

#### 1. Arduino IDE 1.8.9

Penggunaan Arduino IDE 1.8.9 untuk melakukan proses menuliskan kode program dan melakukan proses *upload* ke *microcontroller* arduino nano. Selain itu karena program yang dibuat menggunakan bahasa C dan *microcontroller* yang digunakan arduino maka pada penelitian ini menggunakan perangkat lunak arduino IDE.

#### 2. Eagle 9.0.1

Penggunaan Eagle 9.0.1 untuk perancangan pembuatan PCB. Pcb di sini dibuat untuk memperingkas dalam proses *wiring* pengkabelan di dalam alat yang dibuat. Selain itu ketika menggunakan pcb akan membuat efisien dalam penempatan sensor dan *microcontroller* yang digunakan sehingga alat yang dibuat menjadi lebih kecil daripada tidak menggunakan pcb. Dengan ukuran alat yang lebih semakin kecil akan mempermudah dalam penggunaan dan mempermudah mobilitas dalam pergerakannya.

#### 3. Android Studio 3.5.1

Penggunaan Android Studio 3.5.1 untuk melakukan proses pembuatan aplikasi *smartphone* berbasis android. Dengan adanya aplikasi *smartphone* di tujuan hasil klasifikasi kualitas air yang didapat langsung bisa diterima oleh pengguna dan informasi tersebut bisa digunakan untuk kegiatan yang lainnya.

#### 4. Google Chrome 77.0.3865.90

Penggunaan Google Chrome 77.0.3865.90 untuk proses pembuatan *database firebase* yang berbasis web dan digunakan untuk melakukan pengecekan data yang dikirim menuju firebase ada kendala atau tidak. Google chrome sendiri merupakan sebuah web browser buatan pihak google yang bisa berjalan di berbagai sistem operasi mulai dari windows, mac os, linux dan android atau ios.

#### 5. *OneWire.h*

*Library OneWire.h* digunakan pada penelitian ini untuk melakukan akuisisi data pada sensor suhu air.

#### 6. *DallasTemperature.h*

*Library DallasTemperature.h* digunakan pada penelitian ini untuk melakukan akuisisi data pada sensor suhu air.

#### 7. *GravityTDS.h*

*Library GravityTDS.h* berfungsi untuk melakukan proses akuisisi data pada sensor TDS yang digunakan pada penelitian ini.



#### 8. *EEPROM.h*

*Library EEPROM.h* digunakan untuk sensor TDS pada penelitian ini. Hal tersebut karena pada sensor TDS terdapat EEPROM untuk menyimpan data kalibrasi sebelum dilakukan pengujian sensor.

### 4.3 Batasan Desain Sistem

Batasan desain sistem yang kami buat pada penelitian ini dengan tujuan agar sistem lebih terfokus ketika implementasi sistem. Batasan batasan dari sistem ini adalah

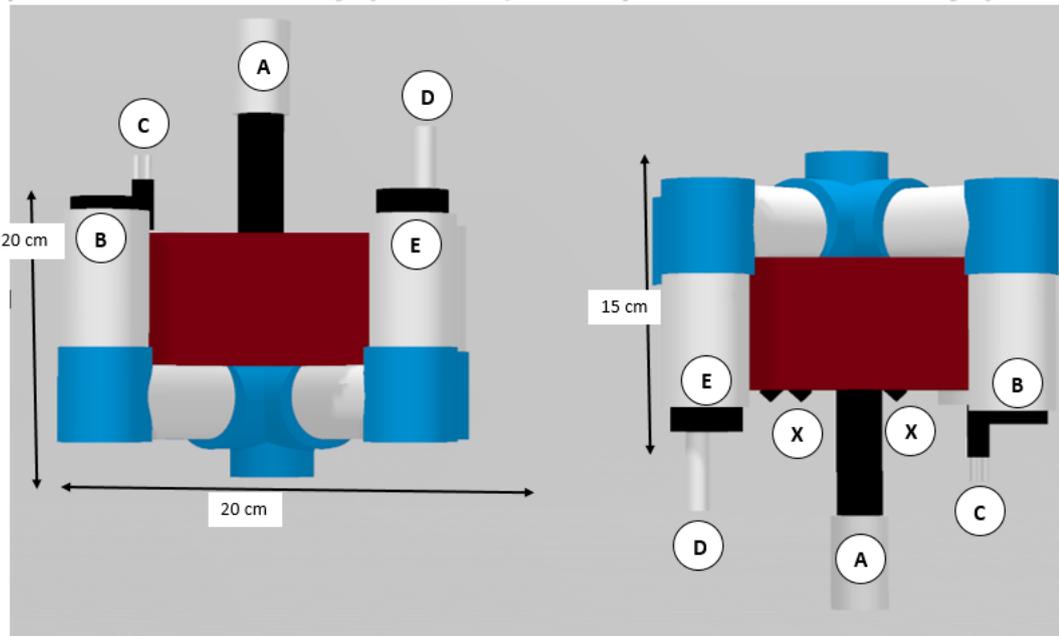
1. Alat yang dibuat menggunakan sensor JSN SR04T yaitu sensor kedalaman air yang memiliki *range* 20 CM – 450 CM sehingga ketika pengujian akan ditempatkan ke lokasi air dengan kedalaman minimal 20 CM.
2. Pada alat yang dibuat menggunakan baterai *lipso* 12V dengan konsumsi daya sebesar 5V untuk arduino, node MCU dan sensor yang digunakan dan 12V untuk motor DC sebagai penggerak alat agar bisa mengelilingi bendungan.
3. Aplikasi yang digunakan berbasis *platform* Android dan memerlukan internet untuk mengakses data yang ada di firebase.
4. Alat yang dibuat memerlukan internet untuk melakukan proses pengiriman hasil klasifikasi menuju firebase yang menjadi *database*.

## BAB 5 PERANCANGAN DAN IMPLEMENTASI

### 5.1 Perancangan Sistem

Perancangan sistem pada sub bab ini akan menjelaskan tentang isi dari perancangan perangkat keras dan perancangan perangkat lunak. Perancangan perangkat keras menjelaskan rangkaian sistem yang akan dibuat agar mudah dipahami. Pada perancangan perangkat lunak menjelaskan diagram alir sistem yang berjalan sehingga bisa dipahami alur sistem berjalan.

#### 5.1.1 Perancangan Purwarupa



Tampak Dari Bawah

Tampak Dari Samping

Gambar 5.1 Perancangan Purwarupa Alat Untuk *Monitoring*

Keterangan :

- A : Sensor pH
- B : Sensor Kedalaman Air
- C : Sensor TDS
- D : Sensor Suhu Air
- E : Sensor Kekeruhan Air
- X : Baling-baling pada purwarupa Alat

Purwarupa alat yang dibuat pada penelitian ini berukuran P = 45 CM, L = 15 CM dan T = 10 CM seperti Gambar 5.1. Bahan untuk pembuat pada purwarupa seperti gambar 5.1 merupakan bahan anti air yang berfungsi agar tidak mudah rusak ketika terkena air. Urutan dari sensor yang dipasang pada alat tersebut yaitu sensor pH diletakkan pada titik tengah alat agar alat tersebut menjadi stabil ketika bergerak mengelilingi bendungan. Lalu untuk 4 sensor lainnya diletakkan pada pojok sisi alat agar alat tersebut stabil dan tidak mudah terbalik ketika berada di atas air. Untuk



perangkat keras lainnya pada sistem ini diletakkan di bagian dalam alat tersebut yang berisi arduino nano yang berfungsi untuk mengakuisisi data sensor, node MCU berfungsi untuk melakukan proses klasifikasi k-NN dan mengirim data ke firebase. Baterai *lippo* digunakan sebagai sumber daya sistem pada penelitian ini.

### 5.1.2 Perancangan Perangkat Keras

#### 1. Perancangan Keseluruhan Sistem

Untuk mengklasifikasikan kualitas air maka sistem ini menggunakan lima sensor yaitu sensor pH untuk mengukur kadar pH air pada bendungan, sensor suhu untuk mengukur suhu air pada bendungan tersebut, sensor kekeruhan air untuk mengukur tingkat kekeruhan air, sensor kedalaman air untuk mengukur seberapa dalam bendungan yang di uji dan sensor TDS untuk mengukur jumlah zat yang terkandung di dalam air di bendungan yang akan diuji. Kelima parameter yang digunakan diharapkan dapat menggambarkan kualitas air yang sebenarnya.

Perancangan seperti Gambar 5.2 dimulai dari arduino nano diberi daya dari 5V *driver* motor melalui pin Vin dan disambungkan pin *ground* antara arduino dengan *driver* motor agar tersambung daya tersebut. Lima sensor yang digunakan semuanya menggunakan daya 5V hasil dari baterai yang di *step down* oleh *driver* motor L298N. Sensor TDS diletakkan pada pin A2 dan *ground* dikarenakan sensor TDS memiliki *output* sensor berupa nilai analog. Sensor *Turbidity* atau sensor kekeruhan air diletakkan di pin A3 dan pin *ground* arduino nano dikarenakan nilai *output* dari sensor *turbidity* berupa nilai analog. Sensor suhu diletakkan di pin digital 4 dan *ground* arduino nano. Sensor pH diletakkan di pin A1 dan *ground* arduino dikarenakan nilai *output* sensor berupa analog. Sensor kedalaman air diletakkan di pin digital 7 untuk pin *echo* dan pin digital 8 untuk pin *trigger* dan *ground* ke arduino nano karena sensor kedalaman air berupa sensor sonar yang berprinsip kerja *transmitter* dan *receiver*. Semua rangkaian sesuai dengan Tabel 5.1 di bawah ini.

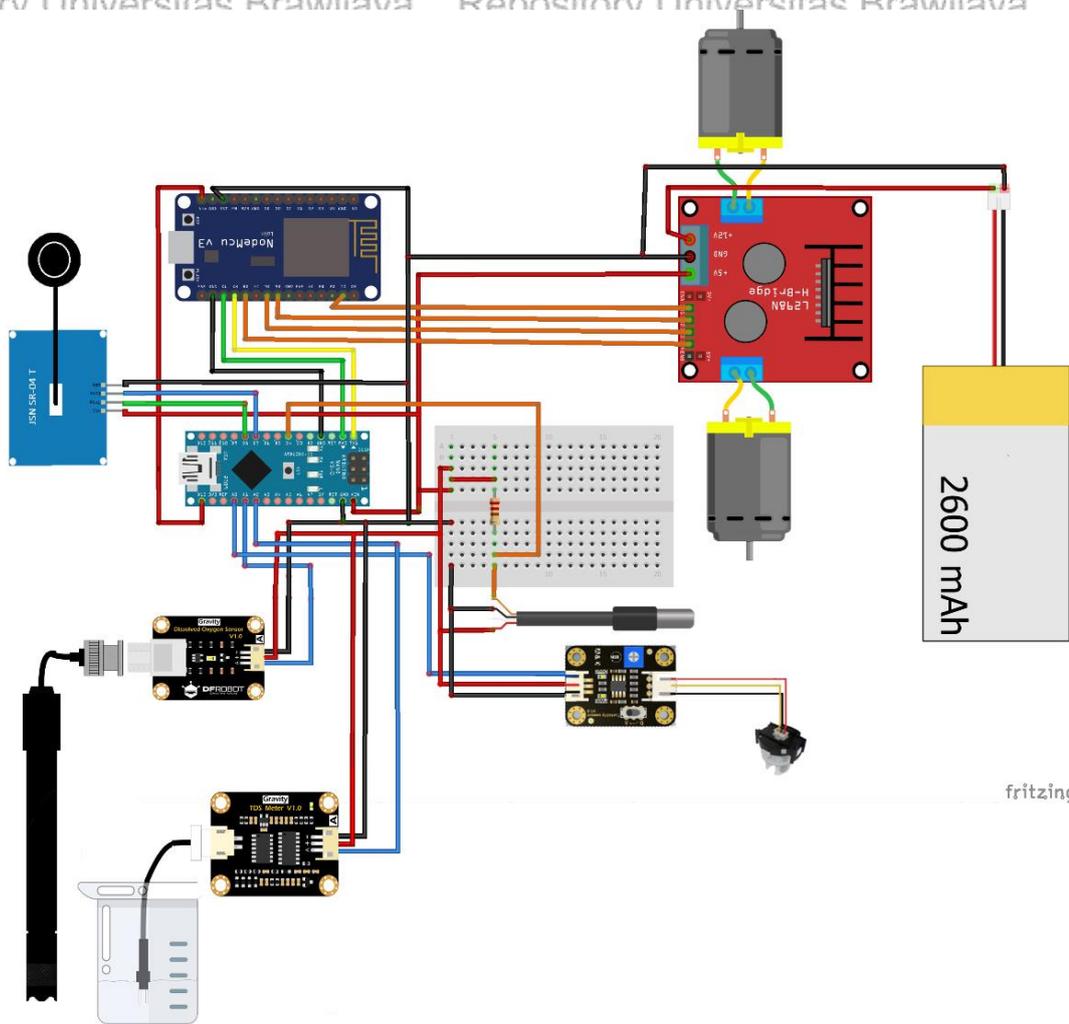
Selain itu terdapat node MCU yang memiliki fungsi utama melakukan proses klasifikasi k-NN dan untuk mengirim data akuisisi sensor dari arduino dan data proses klasifikasi k-NN menuju *database*. Node MCU diberi daya 5V dari *driver* motor dan disambungkan *ground* nya, selain itu fungsi node MCU untuk mengatur PWM 2 motor DC sebagai penggerak alat ketika di atas air. Node MCU memerlukan koneksi internet untuk meminta data dari firebase untuk menggerakkan alat tersebut sesuai data yang dikirim oleh aplikasi *smartphone* pengguna. Semua perancangan rangkaian untuk node MCU sesuai dengan Tabel 5.2 di bawah ini.

**Tabel 5.1 Skema Pin Arduino Nano**

Pin Arduino Nano	Deskripsi
Pin Vin	Pin Vin arduino terhubung dengan 5V <i>driver</i> motor sebagai sumber daya arduino nano.
Pin <i>ground</i>	Pin <i>ground</i> arduino terhubung dengan <i>ground driver</i> motor agar bisa terhubung daya nya.



Pin A1 Analog	Pin A1 terhubung dengan sensor pH Air
Pin A2 Analog	Pin A2 terhubung dengan sensor TDS
Pin A3 Analog	Pin A3 terhubung dengan sensor Kekeruhan
Pin 4 Digital	Pin D4 terhubung dengan sensor suhu air (DS18B20)
Pin 7 Digital	Pin D7 terhubung dengan sensor kedalaman air <i>Echo</i> (JSN-SR04T)
Pin 8 Digital	Pin D8 terhubung dengan sensor kedalaman air <i>Trigger</i> (JSN-SR04T)
Pin RX	Pin RX terhubung dengan pin TX node MCU
Pin TX	Pin TX terhubung dengan pin RX node MCU
Pin <i>ground</i>	Pin <i>ground</i> terhubung dengan pin <i>ground</i> node MCU



Gambar 5.2 Perancangan Perangkat Keras Keseluruhan Sistem

**Tabel 5.2 Skema Pin Node MCU**

Pin Node MCU	Deskripsi
Pin Vin	Pin Vin terhubung dengan 5V dari <i>driver</i> motor sebagai sumber daya node MCU
Pin <i>ground</i>	Pin <i>ground</i> node MCU terhubung dengan <i>ground driver</i> motor agar bisa terhubung daya nya.
Pin RX	Pin RX terhubung dengan pin TX Arduino Nano
Pin TX	Pin TX terhubung dengan pin RX Arduino Nano
Pin <i>ground</i>	Pin <i>ground</i> terhubung dengan pin <i>ground</i> node MCU
Pin D2 Digital	Pin D2 digital node MCU terhubung dengan <i>driver</i> motor sebagai pengatur PWM motor
Pin D5 Digital	Pin D5 digital node MCU terhubung dengan <i>driver</i> motor sebagai pengatur PWM motor
Pin D6 Digital	Pin D6 digital node MCU terhubung dengan <i>driver</i> motor sebagai pengatur PWM motor
Pin D8 Digital	Pin D8 digital node MCU terhubung dengan <i>driver</i> motor sebagai pengatur PWM motor

**2. Perancangan Skema Sensor pH-014**

Perancangan skema sensor pH-014 seperti pada Gambar 5.4 di bawah ini dengan terdapat sensor pH-014, arduino nano, *driver* motor L298N dan baterai *lippo*. Sensor pH di sini difungsikan sebagai *input* data dari sensor berupa pH air yang terdeteksi dan akan diterima dan diproses oleh arduino nano sebagai *central prosesingnya*. Daya untuk sensor pH dan arduino nano yang digunakan berasal dari pin 5V pada *driver* motor L298N. Pin *Vout* pada sensor pH dihubungkan dengan pin Analog A1 di arduino nano karena sensor pH memiliki *output* data analog sehingga perlu disambungkan ke arduino pada pin analog, pin *VCC* sensor pH dihubungkan dengan *output* 5V dari *driver* motor dan pin *ground* nya dihubungkan dengan pin *ground* arduino nano sesuai dengan Tabel 5.3 dan Tabel 5.4. Arduino nano sendiri diberi daya dari baterai dengan pin Vin nya terhubung dengan pin 5V *driver* motor dan pin *ground* arduino nano terhubung dengan pin *ground driver* motor.

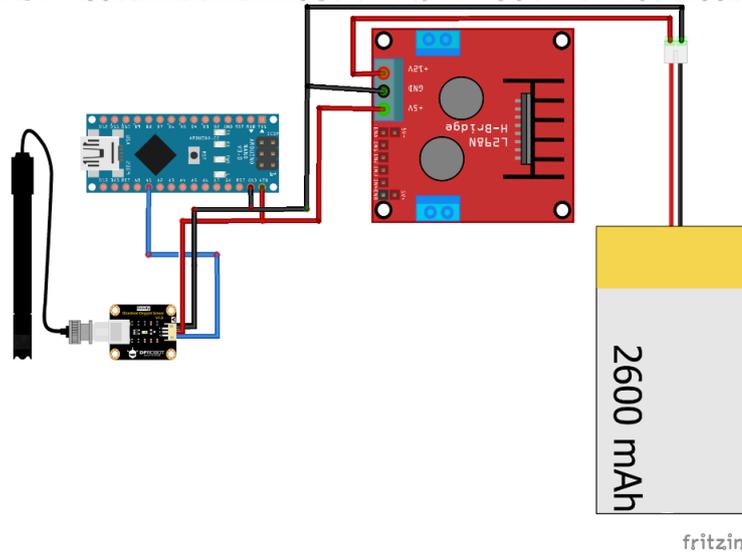
**Tabel 5.3 Tabel Skema Pin Sensor pH dengan Arduino Nano**

Pin Sensor Ph-014	Pin Arduino Nano
<i>ground</i>	<i>ground</i>
<i>Vout</i>	Analog A1



Tabel 5.4 Tabel Skema Pin Sensor pH dengan Driver Motor

Pin Sensor Ph-014	Pin Driver Motor
VCC	5V



Gambar 5.3 Perancangan Skema Sensor pH-014

### 3. Perancangan Skema Sensor Suhu DS18B20

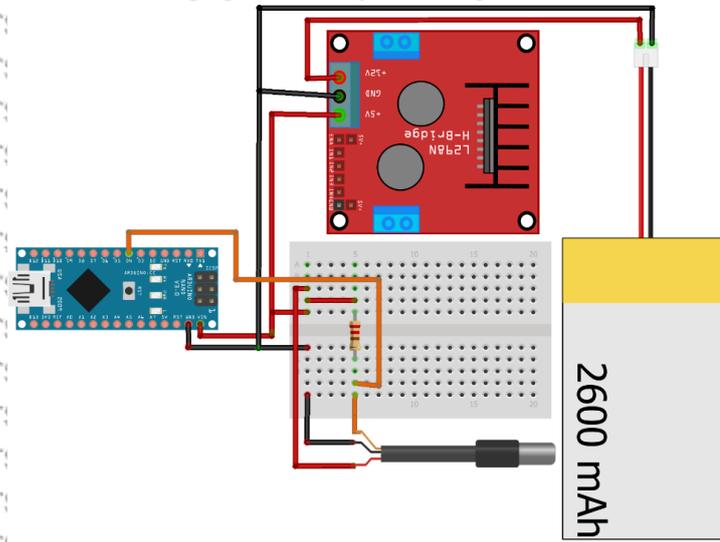
Perancangan skema sensor suhu DS18B20 seperti pada Gambar 5.5 di bawah ini dengan terdapat sensor suhu DS18B20, arduino nano, Driver motor L298N dan baterai Lippo. Sensor suhu DS18B20 di sini difungsikan sebagai *input* data dari sensor berupa suhu air yang terdeteksi dan akan diterima dan diproses oleh arduino nano sebagai *central prosesingnya*. Daya untuk sensor suhu DS18B20 dan arduino nano berasal dari pin 5V driver motor L298N. Pin *Vout* pada sensor suhu DS18B20 dihubungkan dengan pin Digital D4 di arduino nano yang dihubungkan dengan resistor 4.7k terlebih dahulu dengan pin 5V terlebih dahulu, pin *vcc* sensor suhu DS18B20 dihubungkan dengan *output* 5V dari driver motor dan pin *ground* nya dihubungkan dengan pin *ground* arduino nano sesuai dengan Tabel 5.5 dan Tabel 5.6. Arduino nano sendiri diberi daya dari baterai dengan pin *Vin* nya terhubung dengan pin 5V driver motor dan pin *ground* arduino nano terhubung dengan pin *ground* driver motor.

Tabel 5.5 Tabel Skema Pin Sensor Suhu DS18B20 dengan Arduino Nano

Pin Sensor Suhu DS18B20	Pin Arduino Nano
<i>ground</i>	<i>ground</i>
<i>Vout</i>	Digital D4

Tabel 5.6 Tabel Skema Pin Sensor Suhu DS18B20 dengan Driver Motor

Pin Sensor Suhu DS18B20	Pin Driver Motor
VCC	5V



Gambar 5.4 Perancangan Skema Sensor Suhu DS18B20

#### 4. Perancangan Skema Sensor Kedalaman JSN-SR04T

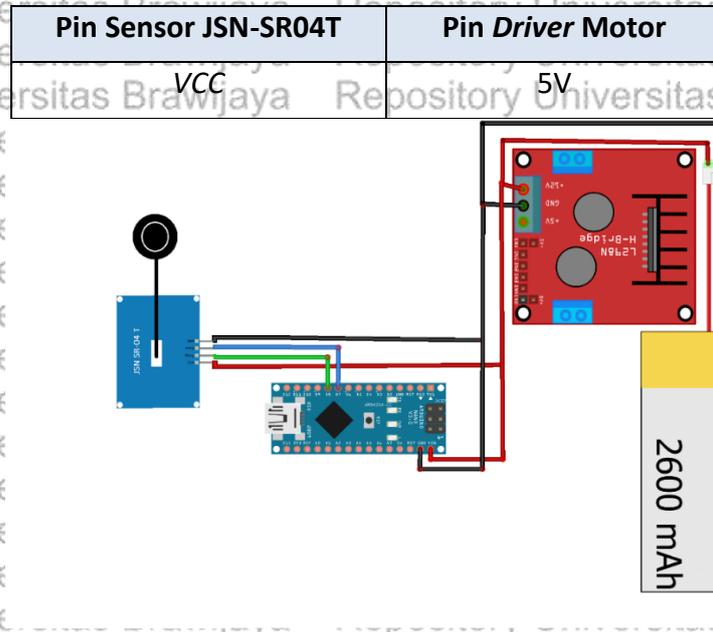
Perancangan skema sensor JSN-SR04T seperti pada Gambar 5.6 di bawah ini dengan terdapat sensor JSN-SR04T, arduino nano, Driver motor L298N dan baterai Lippo. Sensor suhu JSN-SR04T di sini difungsikan sebagai *input* data dari sensor berupa Kedalaman air yang terdeteksi dan akan diterima dan diproses oleh arduino nano sebagai *central prosesingnya*. Daya untuk sensor JSN-SR04T dan arduino nano berasal dari pin 5V driver motor L298N. Pin *Echo* pada sensor JSN-SR04T dihubungkan dengan pin Digital D7 di arduino nano dan pin *trigger* dihubungkan dengan pin Digital D8, pin vcc sensor JSN-SR04T dihubungkan dengan *output* 5V dari driver motor dan pin *ground* nya dihubungkan dengan pin *ground* arduino nano sesuai dengan Tabel 5.7 dan Tabel 5.8. Arduino nano sendiri diberi daya dari baterai dengan pin Vin nya terhubung dengan pin 5V driver motor dan pin *ground* arduino nano terhubung dengan pin *ground* driver motor.

Tabel 5.7 Tabel Skema Pin Sensor JSN-SR04T dengan Arduino Nano

Pin Sensor JSN-SR04T	Pin Arduino Nano
ground	ground
ECHO	Digital D7
TRIGGER	Digital D8



**Tabel 5.8 Tabel Skema Pin Sensor JSN-SR04T dengan Driver Motor**



**Gambar 5.5 Perancangan Skema Sensor Kedalaman JSN-SR04T**

### 5. Perancangan Skema Sensor Kekeruhan Turbidity

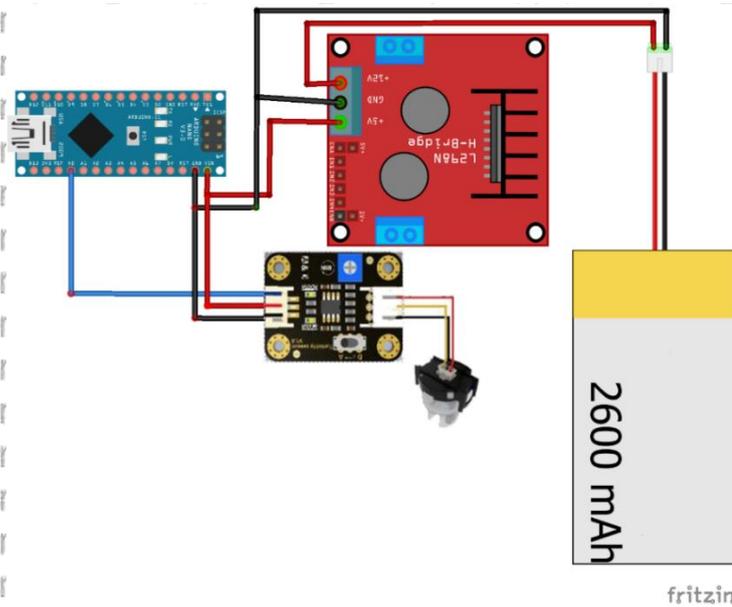
Perancangan skema sensor *Turbidity* seperti pada Gambar 5.6 di bawah ini dengan terdapat sensor *Turbidity*, arduino nano, *Driver* motor L298N dan baterai *Lippo*. Sensor *turbidity* di sini difungsikan sebagai *input* data dari sensor berupa Kekeruhan air yang terdeteksi dan akan diterima dan diproses oleh arduino nano sebagai *central prosesingnya*. Daya untuk sensor *turbidity* dan arduino nano berasal dari pin 5V *driver* motor L298N. Pin *Vout* pada sensor *turbidity* dihubungkan dengan pin Analog A3 di arduino nano, pin *VCC* sensor *turbidity* dihubungkan dengan *output* 5V dari *driver* motor dan pin *ground* nya dihubungkan dengan pin *ground* arduino nano sesuai dengan Tabel 5.9 dan Tabel 5.10. Arduino nano sendiri diberi daya dari baterai dengan pin *Vin* nya terhubung dengan pin 5V *driver* motor dan pin *ground* arduino nano terhubung dengan pin *ground driver* motor.

**Tabel 5.9 Tabel Skema Pin Sensor Turbidity dengan Arduino Nano**

Pin Sensor Turbidity	Pin Arduino Nano
<i>ground</i>	<i>ground</i>
Vout	Analog A3

**Tabel 5.10 Tabel Skema Pin Sensor Turbidity dengan Driver Motor**

Pin Sensor Turbidity	Pin Driver Motor
VCC	5V



Gambar 5.6 Perancangan Skema Sensor Kekerusuhan Turbidity

6. Perancangan Skema Sensor TDS

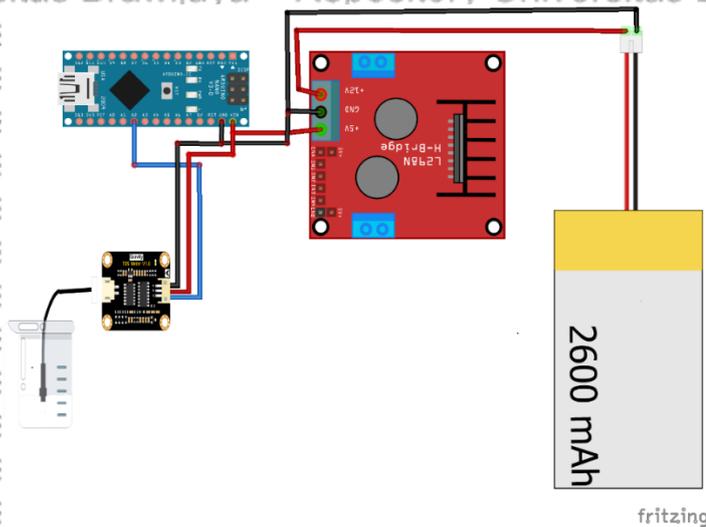
Perancangan skema sensor TDS seperti pada Gambar 5.7 di bawah ini dengan terdapat sensor TDS, arduino nano, Driver motor L298N dan baterai Lippo. Sensor TDS di sini difungsikan sebagai *input* data dari sensor berupa TDS air yang terdeteksi dan akan diterima dan diproses oleh arduino nano sebagai *central prosesingnya*. Daya untuk sensor TDS dan arduino nano berasal dari pin 5V driver motor L298N. Pin Vout pada sensor TDS dihubungkan dengan pin Analog A2 di arduino nano, pin VCC sensor TDS dihubungkan dengan output 5V dari driver motor dan pin ground nya dihubungkan dengan pin ground arduino nano sesuai dengan Tabel 5.11 dan Tabel 5.12. Arduino nano sendiri diberi daya dari baterai dengan pin Vin nya terhubung dengan pin 5V driver motor dan pin ground arduino nano terhubung dengan pin ground driver motor.

Tabel 5.11 Tabel Skema Pin Sensor TDS dengan Arduino Nano

Pin Sensor TDS	Pin Arduino Nano
ground	ground
Vout	Analog A2

Tabel 5.12 Tabel Skema Pin Sensor TDS dengan Driver Motor

Pin Sensor TDS	Pin Driver Motor
VCC	5V



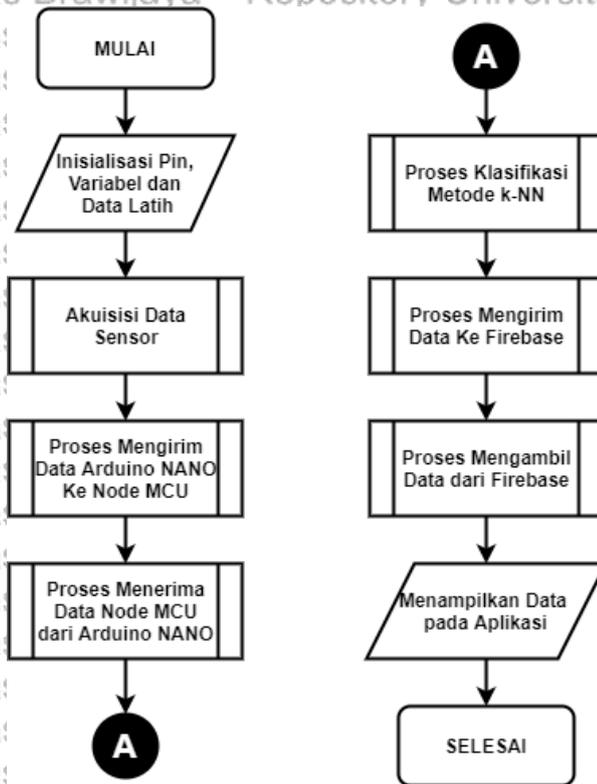
Gambar 5.7 Perancangan Skema Sensor TDS

### 5.1.3 Perancangan Perangkat Lunak

Perancangan perangkat lunak dibagi menjadi beberapa bagian yang dimulai dari perancangan klasifikasi dengan metode k-NN dari *input* lima sensor yang digunakan. Setelah didapat hasil klasifikasi k-NN maka dilanjutkan dengan perancangan pengiriman data dari arduino nano menuju node MCU dan data tersebut akan dikirim ke *database* firebase. Perancangan lainnya berupa perancangan dalam proses menampilkan data yang berada di *database* menuju aplikasi pengguna agar diketahui hasil klasifikasinya.

#### 1. Perancangan Keseluruhan Sistem

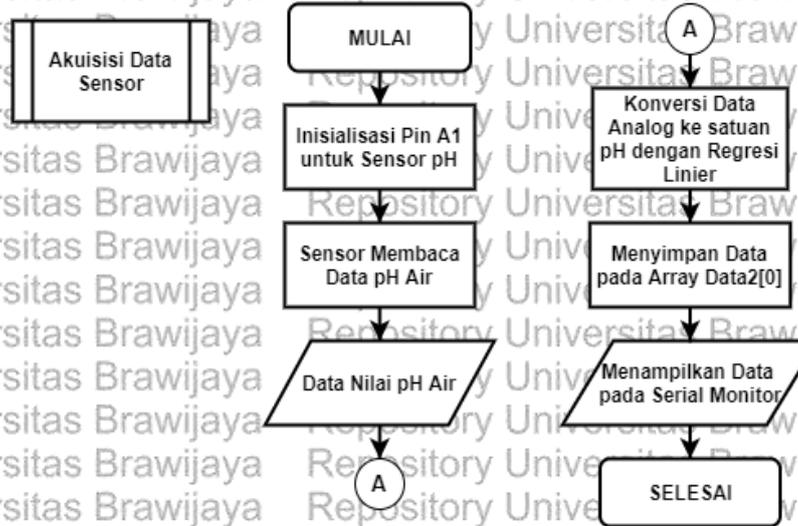
Perancangan perangkat lunak dari keseluruhan sistem pada penelitian ini berbentuk diagram alir seperti Gambar 5.8 di bawah ini. Perancangan tersebut berisi beberapa sub-perancangan dari program keseluruhan sistem yang dibuat. Perancangan dimulai dengan melakukan inisialisasi variabel, pin sensor dan data latih untuk proses selanjutnya. Setelah itu program utama akan masuk ke dalam sub-perancangan akuisisi data sensor. Proses selanjutnya melakukan proses pengiriman data arduino nano ke node MCU yang menjadi sub-perancangan. Ketika data arduino nano diterima oleh node MCU menjadi sub-perancangan maka proses sub-perancangan selanjutnya adalah sub-perancangan klasifikasi k-NN. Pada sub-perancangan ini node MCU sebagai *microcontroller* akan melakukan proses klasifikasi dengan data uji yaitu data 5 sensor dari arduino nano. Setelah proses klasifikasi k-NN selesai maka akan masuk ke sub-perancangan mengirim data ke firebase. Data yang dikirim adalah data 5 sensor dan data klasifikasi k-NN. Setelah itu masuk ke sub-perancangan mengambil data dari firebase dan hasil terakhir dari sistem ini adalah menampilkan data 5 sensor dan hasil klasifikasi k-NN pada aplikasi pengguna.



Gambar 5.8 Perancangan Keseluruhan Sistem

## 2. Perancangan Fungsi Deteksi Sensor pH-014

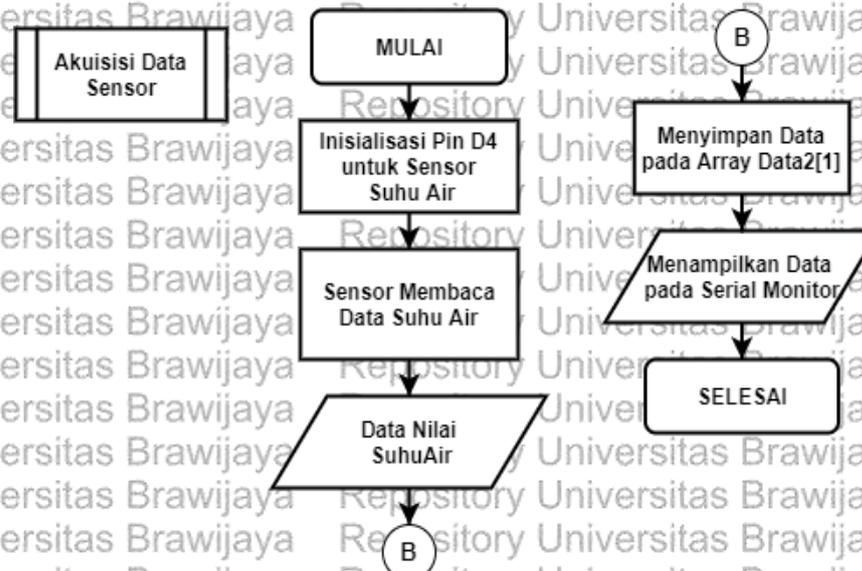
Perancangan untuk deteksi sensor pH-014 seperti pada Gambar 5.9 dimulai dengan melakukan inisialisasi pin A1. Proses selanjutnya adalah melakukan akuisisi data pH air dan data tersebut akan dikonversi ke satuan pH air dengan rumus regresi linier. Proses mendapatkan nilai pada rumus regresi linier dengan cara melihat *output* sensor pH ketika di uji dengan air pH 4, 7, 9 lalu data *output* yang berupa *voltase* tersebut di masukkan ke kalkulator regresi linier online. *Output* sensor pH yang digunakan pada penelitian ini untuk pH 4 sebesar 2,26V, untuk pH 7 *output* nya sebesar 3,35V, dan untuk pH 9 *output* sensor nya adalah 4,42. Untuk *output* tersebut akan memiliki nilai *voltase* yang berbeda-beda di tiap sensor pH yang digunakan. *Output* *voltase* dari sensor pH akan di kalkulasi dengan kalkulator regresi linier maka akan mendapatkan hasil nilai rumus regresi linier, dan pada penelitian rumus regresi linier untuk sensor pH yang didapatkan sebesar  $\text{Nilai pH} = (-3.250 * \text{dataTegangan}) + 3.636$ . Untuk nilai hasil regresi linier tersebut akan tergantung dari data yang dimasukkan ke kalkulator regresi linier dan ketika semakin banyak data yang dimasukkan ke kalkulator regresi linier maka nilai hasil regresi linier akan menjadi lebih baik. Ketika proses konversi nilai pH menggunakan rumus regresi linier sudah selesai maka data akan di masukkan ke dalam *array* data2[0] untuk dilakukan proses selanjutnya pada sistem. *Output* dapat dilihat juga pada serial monitor arduino IDE.



Gambar 5.9 Perancangan Fungsi Deteksi Sensor pH-014

### 3. Perancangan Fungsi Deteksi Sensor Suhu Air

Perancangan sensor suhu untuk mendeteksi suhu air seperti pada Gambar 5.11. Perancangan sensor suhu dimulai dengan inisialisasi pin D4 untuk sensor suhu. Jenis sensor suhu yang digunakan adalah sensor suhu DS18B20. Sensor suhu yang digunakan memiliki *output* data berupa data digital sehingga perlu diberi ke pin digital arduino nano. Setelah itu proses selanjutnya dari sensor suhu akan melakukan pembacaan suhu air. Ketika proses pembacaan sudah selesai maka data tersebut akan dimasukkan ke *array* Data[1] untuk proses selanjutnya. Data[1] tersebut akan dikirim oleh arduino nano ke node MCU untuk menjadi salah satu parameter klasifikasi metode k-NN dan data sensor suhu akan dikirim oleh node MCU ke firebase agar dapat ditampilkan di aplikasi pengguna.

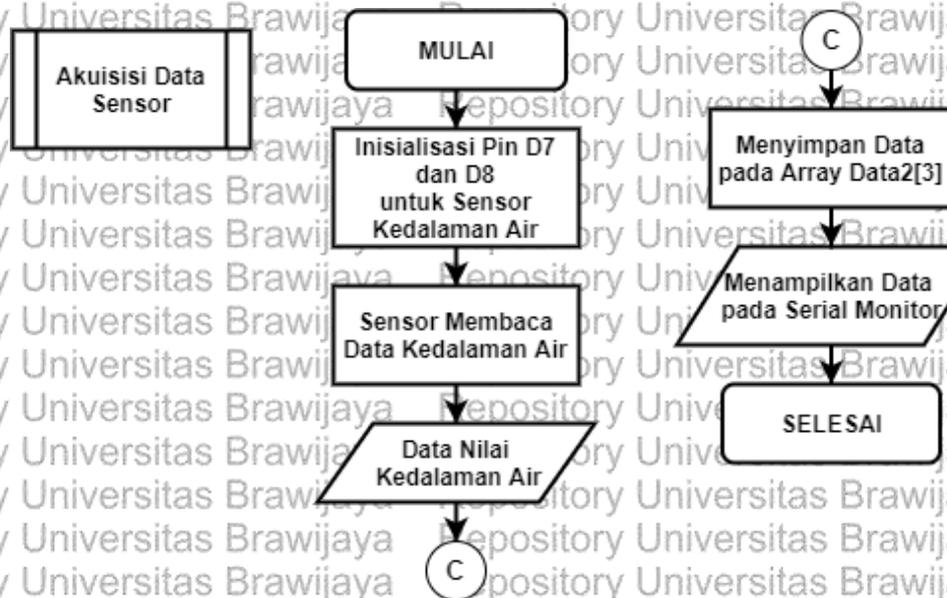


Gambar 5.10 Perancangan Fungsi Deteksi Sensor Suhu Air



#### 4. Perancangan Fungsi Deteksi Sensor Kedalaman Air

Sensor kedalaman air memiliki beberapa proses dalam akuisisi data. Perancangan dimulai dari proses inisialisasi pin D7 dan D8. *Output* dari sensor kedalaman air berupa digital dan terdapat 2 pin yaitu *trigger* dan *echo*. Pin D7 digunakan untuk pin *trigger* sensor dan untuk pin D8 digunakan untuk pin *echo* dari sensor. Proses selanjutnya seperti pada Gambar 5.11 adalah melakukan proses membaca kedalaman air bendungan. Setelah proses tersebut maka akan masuk ke dalam proses menyimpan data kedalaman air ke dalam sebuah *array* Data2[2]. *Array* tersebut akan digunakan untuk proses selanjutnya seperti untuk proses klasifikasi k-NN oleh node MCU dan untuk dikirim ke firebase. *Index array* untuk data sensor kedalaman air adalah *index* ke 2 dan *index* tersebut digunakan untuk mempermudah dalam proses memilah data pada *array* Data2[*i*].

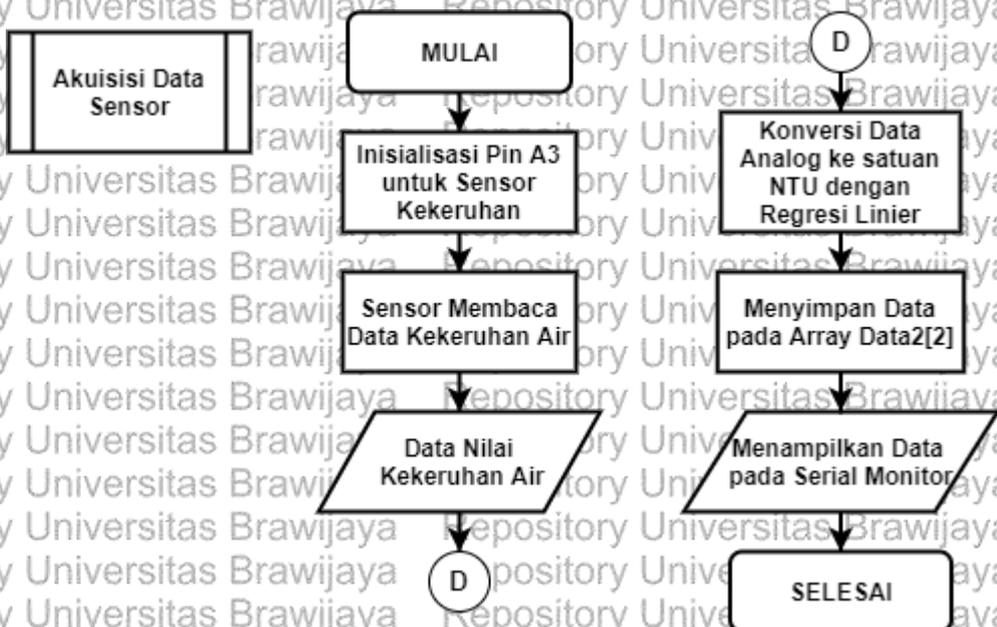


Gambar 5.11 Perancangan Fungsi Deteksi Sensor Kedalaman Air

#### 5. Perancangan Fungsi Deteksi Sensor Kekeruhan Air

Sensor kekeruhan air memiliki *output* data berupa analog voltase. Proses sub-perancangan pembacaan data sensor kekeruhan sesuai dengan Gambar 5.12 dimulai dari inisialisasi pin A3 pada arduino nano. Proses selanjutnya adalah melakukan pembacaan kekeruhan air yang diuji. Data yang diterima arduino nano berupa analog voltase dan satuan kekeruhan adalah NTU. Proses selanjutnya adalah konversi data analog voltase menjadi satuan NTU dengan rumus regresi linier. Rumus regresi linier didapatkan dimulai dengan melihat *output* sensor kekeruhan yang berupa voltase ketika proses kalibrasi dengan air yang memiliki tingkat kekeruhan yang berbeda beda. Pada penelitian ini proses pengujian pengambilan *output* untuk dimasukkan ke rumus regresi linier yaitu dengan menggunakan air yang berjumlah 5 sampel dengan tingkat kekeruhan yang berbeda beda, dimulai dengan air bersih hingga air yang paling keruh. Setelah mendapatkan nilai *output* sensor kekeruhan dengan 5 sampel air

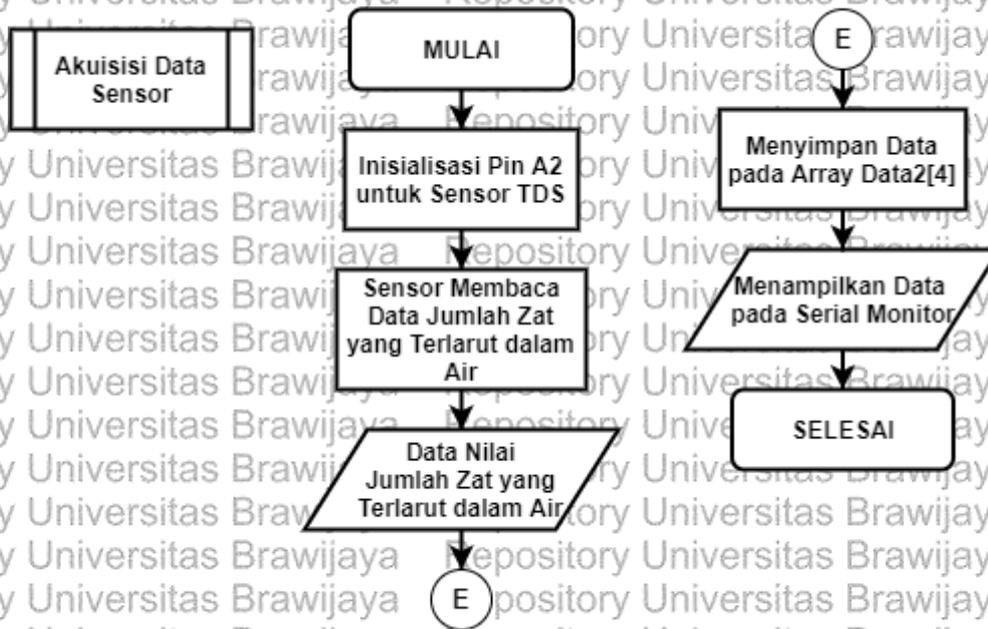
tersebut maka nilai voltase tersebut akan dimasukkan ke kalkulator regresi linier online. Hasil rumus yang didapatkan dari kalkulator regresi linier untuk penelitian ini adalah  $ntu = (- 65.20 * volt ) + 280$ . Nilai tersebut akan berbeda sesuai *output* sensor kekeruhan yang digunakan dan jumlah air sampel yang di uji untuk datanya dimasukkan ke kalkulator regresi linier. Hasil dari sensor yang sudah berupa satuan NTU akan disimpan pada *array* Data2[2] pada arduino nano. *Index* ke 3 untuk menyimpan data kekeruhan pada *array* tersebut bertujuan untuk mempermudah dalam proses selanjutnya ketika proses klasifikasi k-NN pada node MCU.



Gambar 5.12 Perancangan Fungsi Deteksi Sensor Kekeruhan Air

### 6. Perancangan Fungsi Deteksi Sensor TDS

Sensor TDS merupakan sensor untuk mengetahui jumlah zat yang terlarut di dalam air. Sensor TDS melakukan proses pembacaan data sesuai dengan Gambar 5.13 dimulai dari inisialisasi pin A2 pada arduino nano. Setelah proses inialisasi pin untuk sensor tds maka akan masuk ke proses selanjutnya yaitu melakukan pembacaan data pada air untuk pengujian. Setelah sensor tds mendapatkan data maka data tersebut akan diproses oleh arduino nano untuk proses selanjutnya. Proses tersebut adalah menyimpan data yang didapatkan sensor tds pada *array* Data2[4]. *Array* Data2[4] akan digunakan untuk proses selanjutnya pada sistem yang dibuat pada penelitian ini.



Gambar 5.13 Perancangan Fungsi Deteksi Sensor TDS.

### 7. Perancangan Proses Mengirim Data Arduino Nano ke Node MCU

Proses pengiriman data dari arduino nano ke node MCU merupakan salah satu sub-perancangan dari perancangan keseluruhan sistem. Proses dimulai dengan proses akuisisi data dari 5 sensor dilakukan oleh arduino nano. Setelah proses akuisisi data dari 5 sensor sudah selesai maka akan masuk ke proses selanjutnya yaitu melakukan proses menyimpan data 5 sensor tersebut menjadi 1 paket data float. Untuk urutan paket data yang akan dikirim ke node MCU adalah sebagai berikut urutan pertama adalah nilai sensor pH yang disimpan pada array Data2[0], lalu data sensor suhu yang disimpan pada array Data2[1], lalu data sensor kedalaman air yang disimpan pada array Data2[2]. Setelah itu data sensor kekeruhan yang sudah disimpan pada array Data[3] dan data sensor tds yang sudah didalam data array Data2[4]. Proses selanjutnya adalah merubah data tersebut dari type float menjadi *String* agar dapat dikirim arduino nano ke node MCU. Setelah data tersebut menjadi string maka proses selanjutnya adalah merubah data string tadi menjadi data char karena para proses pengiriman data menggunakan komunikasi serial data yang bisa dikirim adalah data dengan typer char. Setelah data bertype char maka data yang berisi 5 data sensor tersebut dapat dikirim menggunakan komunikasi serial. Komunikasi serial merupakan pengiriman data dari satu device ke device lainnya menggunakan pin RX TX. Proses selanjutnya adalah proses mengirim data tersebut ke node MCU melalui pin RX TX yang sudah terhubung. Proses pengiriman tersebut menggunakan metode komunikasi UART (*Universal Asynchronous Receiver-Transmitter*). Metode komunikasi tersebut biasanya digunakan untuk komunikasi serial antar perangkat dengan mengirim data per bit-bit dan proses menerjemahkan bit-bit yang terpecah tersebut menjadi satu lagi.



Proses Mengirim Data Arduino NANO Ke Node MCU

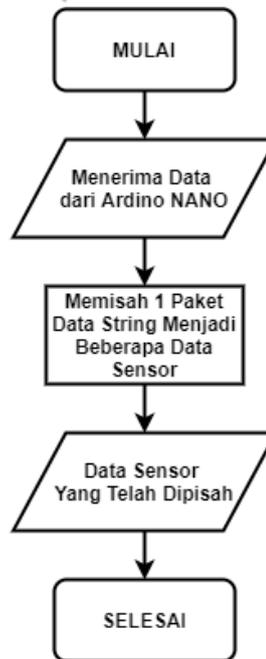


Gambar 5.14 Perancangan Pengiriman Data Arduino ke Node MCU

**8. Perancangan Proses Menerima Data Node MCU dari Arduino Nano**

Proses menerima data pada node MCU dari arduino nano menggunakan komunikasi UART. Proses dimulai dengan menerima data tiap bit-bit paralel dan menggabungkan menjadi satu. Data yang dikirim berisi 5 data yaitu data sensor pH, data sensor Kekeruhan, data sensor TDS, data sensor suhu dan data sensor kedalaman air. Proses untuk menerima data dari node MCU seperti pada Gambar 5.15 di bawah ini.

Proses Menerima Data Node MCU dari Arduino NANO



Gambar 5.15 Perancangan Proses Menerima Data Node MCU dari Arduino



## 9. Perancangan Klasifikasi k-NN

Sistem yang dibuat pada penelitian ini dimulai dari perancangan klasifikasi metode k-NN yang menjadi fokus penelitian ini. Dalam klasifikasi k-NN dimulai dari penentuan jumlah K atau tetangga terdekat hasil perhitungan data uji dengan data latih. Penentuan nilai K sangatlah mempengaruhi nilai akurasi dari klasifikasi metode k-NN. Setelah penentuan jumlah K maka selanjutnya sistem akan melakukan akuisisi sensor yang berjumlah lima untuk diambil data tersebut untuk digunakan data *input* klasifikasi. Data hasil akuisisi tersebut akan dilakukan normalisasi karena untuk keseimbangan *range* data yang akan dimasukkan ke rumus *euclidean*. Normalisasi yang dipilih pada penelitian ini adalah normalisasi min-max. Ketika data 5 sensor setelah di normalisasi min-max maka akan memiliki range data 0 hingga 1 dan memiliki keseimbangan *range* data. Setelah proses normalisasi maka data sensor tersebut akan dihitung dengan rumus *euclidean* dengan setiap data latih yang tersedia untuk menentukan jarak tetangga data uji dengan data latih. Ketika sudah mendapatkan hasil perhitungan dari rumus *euclidean* maka dilakukan pengurutan data dari yang terkecil hingga terbesar. Jika proses sudah selesai sampai ini maka jumlah K akan berperan penting dengan akurasi hasil klasifikasi karena nilai K akan menentukan jumlah data berapa yang akan diambil untuk diproses selanjutnya. Jumlah data sesuai nilai K akan dihitung dengan metode *weighted voting* agar diketahui bobot jarak dari data uji dengan setiap data latih. Hasil dari metode *Weighted voting* akan menjadi hasil akhir klasifikasi metode k-NN. Ketika hasil klasifikasi memiliki akurasi yang kurang maka bisa masuk kembali ke proses penentuan nilai K agar didapat akurasi yang lebih baik lagi (Yolanda, SD, 2019). Ketika hasil akurasi sudah baik maka hasil tersebut akan dikirim menuju node MCU agar data tersebut bisa dikirim ke *database* dan bisa diakses oleh aplikasi *smartphone*. Data yang dikirim ke node MCU berupa lima data dari tiap akuisisi sensor dan hasil klasifikasi metode k-NN. Perancangan Klasifikasi k-NN bisa dilihat seperti Gambar 5.16 di bawah ini.

Proses perancangan proses klasifikasi metode k-NN sebagai berikut :

1. Persiapkan data latih karena metode k-NN merupakan metode *supervised machine learning*.
2. Tentukan jumlah K.
3. Proses normalisasi min-max dari 5 data sensor yang digunakan.
4. Hitung data uji baru yang didapat dari lima sensor yang digunakan dalam alat ini dengan data latih yang sudah disiapkan dengan rumus *Euclidean*.
5. Berikut persamaan 2.1 dari *euclidean* :

$$D = \sqrt{\sum(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2 + (x_4 - y_4)^2 + (x_5 - y_5)^2} \quad (2.1)$$

Keterangan :

D adalah Hasil jarak data latih dengan data uji yang didapatkan oleh kelima sensor.

$x_1$  adalah Data Uji sensor pH air

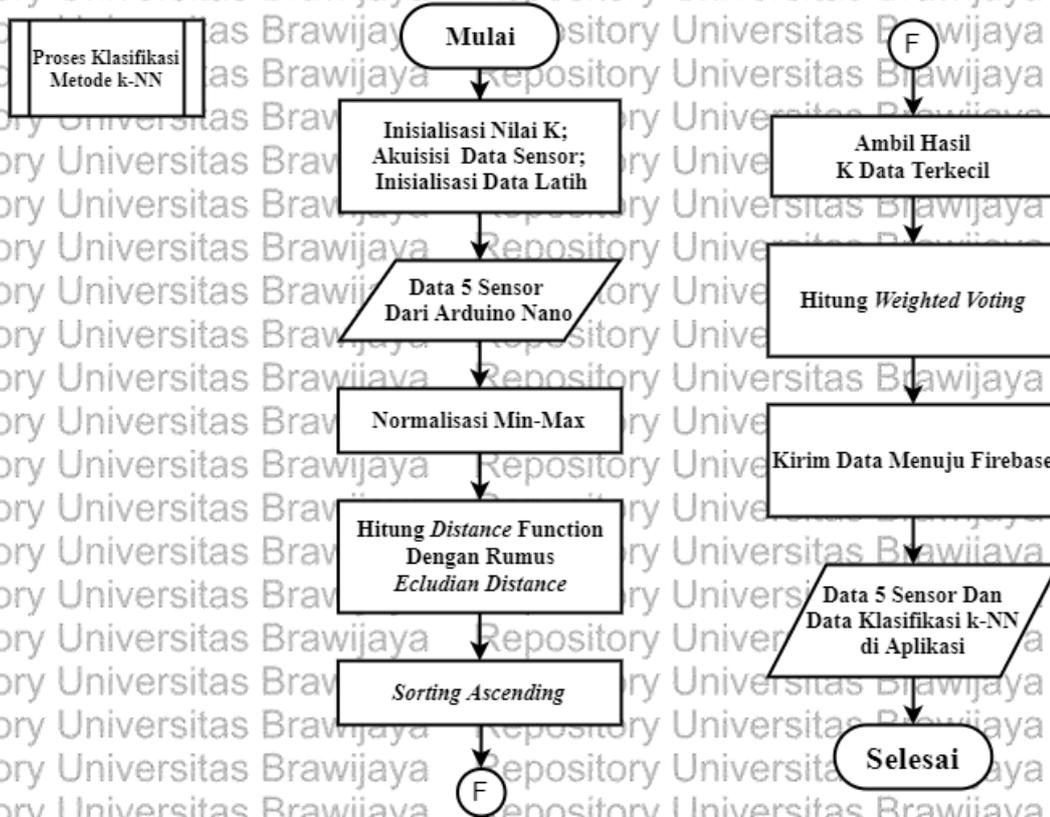
$y_1$  adalah Data Latih sensor pH air.



- $x_2$  adalah Data Uji sensor suhu air
- $y_2$  adalah Data Latih sensor suhu air.
- $x_3$  adalah Data Uji sensor kedalaman air
- $y_3$  adalah Data Latih sensor kedalaman air.
- $x_4$  adalah Data Uji sensor kekeruhan air
- $y_4$  adalah Data Latih sensor kekeruhan air.
- $x_5$  adalah Data Uji sensor TDS
- $y_5$  adalah Data Latih sensor TDS.

6. Hasil perhitungan dari data lima sensor sebagai data uji dengan data latih maka perlu dilakukan *sorting* dari terkecil hingga terbesar (*Ascending*)
7. Setelah data hasil perhitungan diurutkan secara *ascending* maka diambil data sesuai dengan nilai K.
8. Hitung dengan metode *weighted voting* dengan persamaan (2.2) seperti berikut :
 
$$\text{Vote (golongan)} = \frac{1}{\text{data } x^2} + \frac{1}{\text{data } x^2} \approx \text{Hasil} \quad (2.2)$$
9. Hasil *output* dari sistem yang dibuat sesuai dengan hasil *vote* golongan tersebut.
10. *Output* yang sudah sesuai akan dikirim arduino ke node MCU yang selanjutnya data tersebut akan dikirim ke *firebase*.

Berikut ini proses perancangan klasifikasi k-NN seperti pada Gambar 5.9.

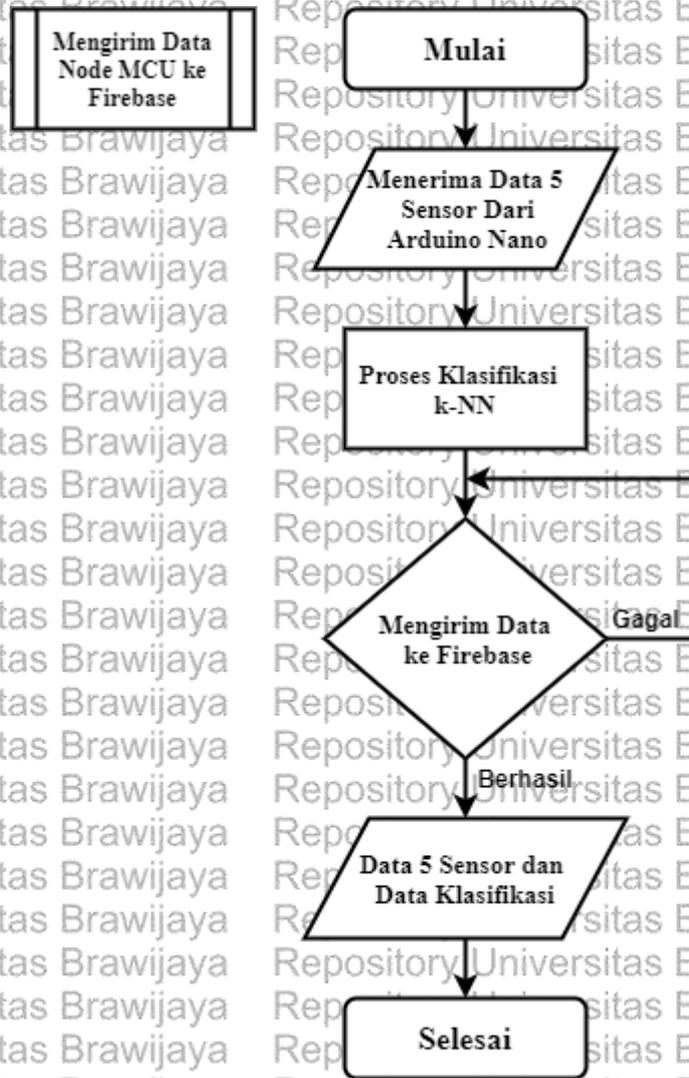


Gambar 5.16 Perancangan Klasifikasi k-NN



### 10. Perancangan Pengiriman Data Node MCU Ke Firebase

Pengiriman data hasil klasifikasi dari arduino NANO akan diproses oleh Node MCU yang digunakan sebagai perantara untuk mengirim menuju database *firebase*. Data hasil akuisisi di arduino nano akan dikirim menuju node MCU dengan menggunakan pin RX TX secara serial. Ketika data sudah diterima node MCU maka node MCU akan bertugas mengirim data tersebut menuju *database firebase* agar data tersebut bisa diakses oleh aplikasi *smartphone*. Sebelum melakukan pengiriman data ke firebase maka node MCU perlu melakukan klasifikasi k-NN dan selanjutnya mengirim data ke firebase yang berisi data 5 sensor dan data hasil klasifikasi k-NN. Perancangan pengiriman data node MCU bisa dilihat seperti Gambar 5.17 di bawah ini.

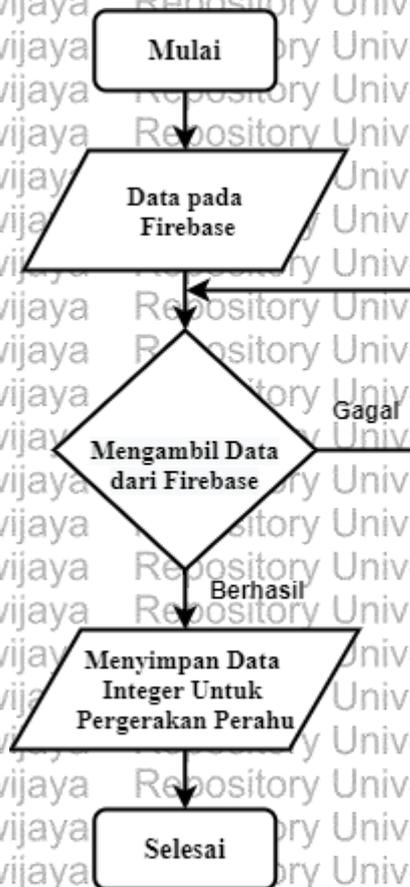


Gambar 5.17 Perancangan Pengiriman data Node MCU

### 11. Perancangan Proses Node MCU Menerima Data dari Aplikasi

Aplikasi *smartphone* dibuat untuk menampilkan data yang dikirim oleh node MCU dan aplikasi *smartphone* dapat berfungsi untuk menggerakkan kapal dengan mengirim data ke firebase. Proses perancangan node MCU menerima

data pada firebase di mulai dengan node MCU menerima data yang berasal dari aplikasi dengan perantara firebase. Data yang diterima oleh node MCU berupa nilai *integer* 0, 1, 2, 3, 4. Proses selanjutnya yang dilakukan oleh node MCU adalah menggerakkan alat sesuai data yang diterima. Perancangan proses menerima data dari aplikasi yang dilakukan oleh node MCU seperti pada Gambar 5.18 di bawah ini.



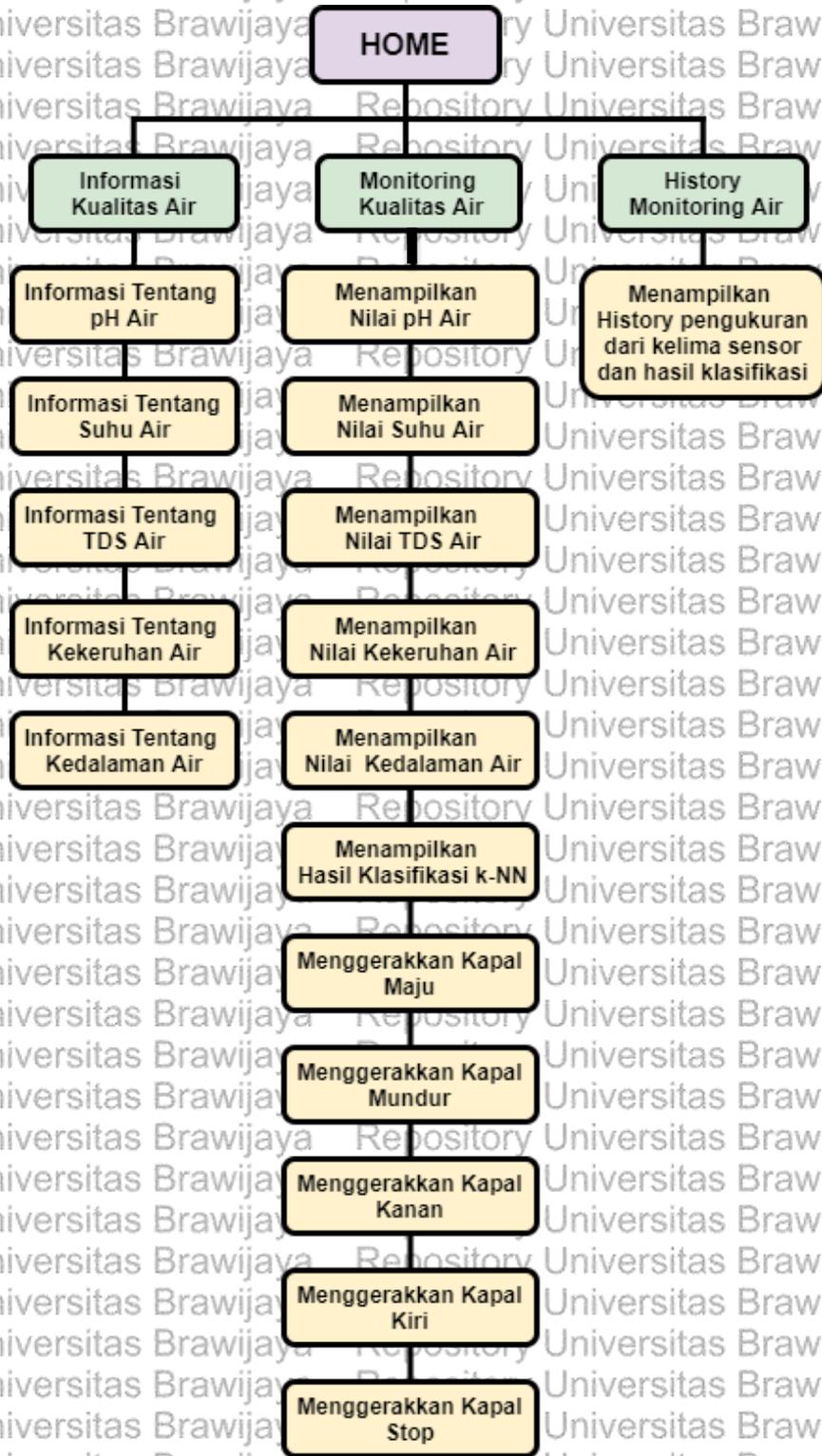
Gambar 5.18 Perancangan Penerimaan Data dari Aplikasi

## 12. Perancangan Proses Pergerakan *Prorotype* Alat

Pergerakan alat pada penelitian ini berdasarkan perintah dari aplikasi *smartphone* pengguna. Setelah node MCU menerima data tersebut melalui firebase maka node MCU akan memproses data tersebut. Data yang berisi 0, 1, 2, 3, 4 memiliki fungsi masing-masing. Ketika node MCU menerima data bernilai 0 maka akan membuat kapal STOP lalu ketika data yang diterima bernilai 1 maka akan membuat kapal bergerak MAJU. Untuk data yang diterima bernilai 2 maka alat akan bergerak MUNDUR, alat akan bergerak ke KIRI ketika menerima data dengan nilai 3 dan alat akan bergerak ke KANAN ketika menerima data dengan nilai 4. Pergerakan alat dikendalikan oleh node MCU menggunakan 4 buah pin *PWM* yang tersedia oleh *microcontroller* node MCU sendiri. Dengan pergerakan alat yang dapat dikendalikan oleh pengguna maka akan mempermudah proses *monitoring* kualitas air dengan purwarupa alat yang dapat mengelilingi bendungan. Perancangan proses penerimaan data dari



sehingga mempermudah pengguna untuk merancang aplikasi yang akan terhubung dengan firebase database.



Gambar 5.20 Sitemap Perancangan Aplikasi



Gambar 5.21 Perancangan Desain Aplikasi Halaman Awal *Smartphone*



Gambar 5.22 Perancangan Tampilan Isi Aplikasi

Tampilan pertama seperti gambar di atas paling kiri merupakan tampilan yang berisi informasi cara mengetahui kualitas air yang baik menurut peraturan pemerintah. Tujuan dari dibuatnya agar pengguna aplikasi bisa mengetahui terlebih dahulu nilai dari tiap parameter yang akan *di monitoring* dengan kualitas yang baik seperti apa sehingga pengguna tidak akan merasa bingung ketika melakukan *monitoring* kualitas air di bendungan.



Tampilan kedua seperti gambar yang berada di tengah tersebut dibuat dengan tujuan pengguna bisa mengetahui hasil dari *monitoring* sensor yang berada di purwarupa alat ketika melakukan pengukuran kualitas air di bendungan. Parameter yang ditampilkan di aplikasi yaitu hasil klasifikasi menggunakan metode k-NN, nilai pH air, nilai suhu air, nilai kekeruhan air, kedalaman air dan jumlah zat terlarut di dalam air. Nilai yang berada di tampilan ini diharapkan bisa menggambarkan kualitas air yang dilihat dari segi parameter fisika. Pada penelitian ini menggunakan *firebase* yang memiliki fitur *realtime database* sehingga data hasil akuisisi dari sensor dapat langsung dilihat di aplikasi si pengguna. Hal tersebut dapat meningkatkan manfaat dari sistem yang dibuat pada penelitian ini. Selain itu terdapat tampilan kontrol pergerakan purwarupa alat yang didesain untuk memudahkan pengguna untuk menggerakkan purwarupa alat tersebut.

Tampilan ketiga merupakan tampilan *history* data dari *monitoring* kualitas air yang telah dilakukan sebelumnya. Tampilan ini dibuat agar hasil pengukuran bisa diakses kemudian hari ketika dibutuhkan untuk kegiatan yang lainnya seperti rekap hasil *monitoring* kualitas air. *History* ini dapat digunakan karena penggunaan *database firebase* yang bisa menyimpan file. File yang disimpan merupakan nilai dari sensor suhu air, pH air, kedalaman air, kekeruhan air, jumlah zat yang terlarut dalam air dan hasil klasifikasi dengan metode k-NN.

#### 14. Perancangan Data Pada Klasifikasi K-NN

Pada penelitian ini menggunakan metode k-NN untuk melakukan klasifikasi kualitas air pada bendungan. Metode k-NN merupakan metode yang memerlukan data latih untuk proses klasifikasi. Pada penelitian ini hasil klasifikasi metode k-NN dibagi menjadi 3 kelas yaitu “Bagus”, “Sedang” dan “Jelek”. Pada penelitian ini menggunakan data dengan jumlah 75 data. Dari 75 data tersebut akan di bagi menjadi 2 bagian yaitu data latih dan data uji. Untuk data latih yang akan digunakan klasifikasi metode k-NN yaitu sejumlah 50 data. Dari 50 data latih tersebut terdiri dari 20 data latih dengan kelas “Bagus”, lalu 20 data dengan label kelas “Sedang” dan 10 data untuk label kelas “Jelek”. Untuk data uji yang berfungsi untuk mengetahui akurasi dari klasifikasi metode k-NN yang digunakan pada penelitian ini adalah sejumlah 25 data. Dari 25 data tersebut terdiri dari 10 data uji dengan label kelas “bagus”, 10 data uji dengan label kelas “Sedang” dan 5 data uji dengan label kelas “Jelek”.

#### 5.2 Implementasi Sistem

Sub bab Implementasi Sistem di dalamnya berisi penjelasan bagaimana cara sistem yang telah dirancang bisa diterapkan secara langsung. Isi dari sub bab implementasi sistem akan dibagi menjadi 3 yaitu implementasi purwarupa, implementasi perangkat keras dan implementasi perangkat lunak.



### 5.2.1 Implementasi Purwarupa

Pada penelitian ini dibuat sebuah purwarupa alat yang bisa melakukan *monitoring* kualitas air di sebuah bendungan. Purwarupa ini bisa mengelilingi bendungan dan bergerak di atas air sesuai dengan perintah di aplikasi *smartphone* pengguna. Bahan dasar dari pembuatan purwarupa alat ini berasal dari bahan anti air agar ketika purwarupa ini melakukan *monitoring* kualitas air, purwarupa tidak mudah rusak terkena air, selain itu pemilihan bahan tersebut ditujukan untuk membuat purwarupa lebih ringan dan lebih awet dalam penggunaannya karena purwarupa ini akan selalu terkena air dalam penggunaannya. Ukuran dari purwarupa seperti pada Gambar 5.22 ini.



**Gambar 5.23 Implementasi Purwarupa Alat Untuk *Monitoring***

- Keterangan :
- A : Sensor pH
  - B : Sensor Kedalaman Air
  - C : Sensor TDS
  - D : Sensor Suhu Air
  - E : Sensor Kekeruhan Air
  - X : Baling-baling pada purwarupa Alat

### 5.2.2 Implementasi Perangkat Keras

#### 1. Implementasi Keseluruhan Sistem

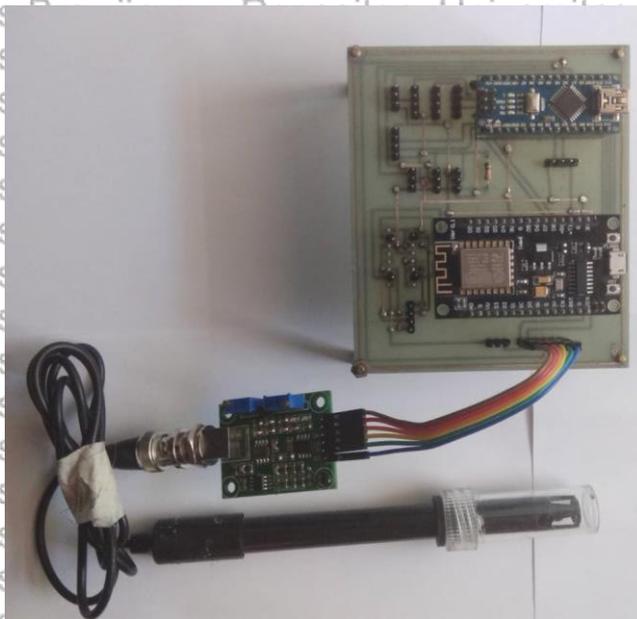
Implementasi keseluruhan sistem dibuat sesuai dengan sub-bab perancangan keseluruhan sistem dengan tujuan untuk mengetahui proses alur kerja dari sistem yang dibuat. Sistem secara keseluruhan dimulai dari arduino nano sebagai salah satu *microcontroller* yang digunakan pada penelitian ini melakukan proses akuisisi data sensor yang digunakan. Sensor yang digunakan adalah sensor pH, sensor suhu, sensor kedalaman air, sensor kekeruhan air dan yang terakhir sensor TDS. Ketika



semua sensor sudah mendapatkan data tentang kualitas air tiap parameter yang diukur maka arduino nano akan mengirim data tersebut ke node MCU. Node MCU digunakan untuk melakukan proses klasifikasi k-NN dan mengirim data ke firebase. Data yang dikirim ke firebase berupa 6 data yaitu data sensor pH, data sensor suhu, data sensor kedalaman air, data sensor kekeruhan air, data sensor TDS dan data hasil klasifikasi metode k-NN. Setelah proses mengirim data maka node MCU akan meminta data ke firebase untuk menggerakkan purwarupa alat sesuai perintah dari aplikasi android si pengguna. 6 data yang dikirim ke firebase akan ditampilkan pada aplikasi android si pengguna untuk melihat data hasil kualitas air dari sistem yang dibuat pada penelitian ini.

## 2. Implementasi Skema Sensor pH-014

Pada implementasi skema sensor pH-014 dibuat sesuai pada bab perancangan di Gambar 5.2. implementasi sensor pH-014 dihubungkan dengan arduino nano yang ditujukan agar nilai dari pembacaan sensor pH-014 dapat dibaca oleh arduino nano sebagai *microcontroller*. Sensor pH-014 dihubungkan dengan arduino menggunakan kabel jumper dengan pin Vin, *ground* dan pin Analog A1 karena nilai dari sensor pH-014 berupa analog. Berikut foto dari implementasi skema sensor pH-014 seperti pada Gambar 5.24.



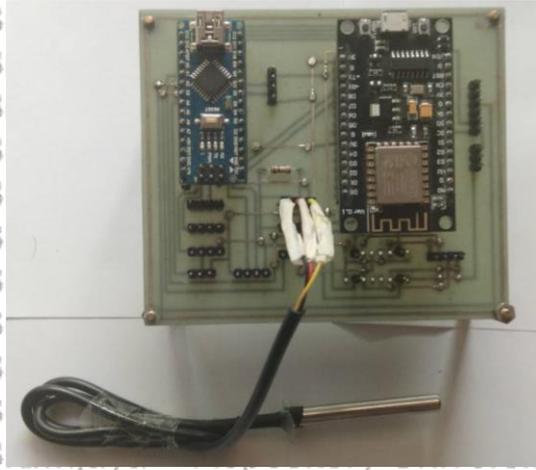
Gambar 5.24 Implementasi Skema Sensor pH-014

## 3. Implementasi Skema Sensor Suhu DS18B20

Pada implementasi skema sensor suhu DS18B20 dibuat sesuai pada bab perancangan di Gambar 5.2. implementasi sensor suhu DS18B20 dihubungkan dengan arduino nano yang ditujukan agar nilai dari pembacaan sensor suhu dapat dibaca oleh arduino nano sebagai *microcontroller* berupa nilai suhu air yang terdeteksi. Pada implementasi ini menggunakan resistor 4,7k yang dihubungkan VCC dengan pin digital D4 dengan tujuan meningkatkan pembacaan sensor suhu tersebut. Sensor suhu DS18B20 dihubungkan dengan arduino menggunakan kabel



jumper dengan pin *Vin*, *ground* dan pin Digital D4. Berikut foto dari implementasi skema sensor suhu DS18B20 seperti pada Gambar 5.25.



Gambar 5.25 Implementasi Skema Sensor Suhu DS18B20

#### 4. Implementasi Skema Sensor Kedalaman JSN-SR04T

Pada implementasi skema sensor kedalaman JSN-SR04T dibuat sesuai pada bab perancangan di Gambar 5.2. sensor kedalaman dihubungkan dengan arduino nano yang ditujukan agar nilai dari pembacaan sensor kedalaman dapat dibaca oleh arduino nano sebagai *microcontroller* berupa nilai kedalaman air yang terdeteksi dengan nilai per centi meter. Sensor kedalaman JSN-SR04T dihubungkan dengan arduino menggunakan kabel jumper dengan pin *Vin*, *ground*, pin Digital D7 (*echo*) dan pin Digital D8 (*trigger*). Sensor kedalaman air diberi daya 5V dari driver motor dan arduino nano juga diberi daya dari driver motor untuk beroperasi sesuai dengan keinginan. Berikut foto dari implementasi skema sensor JSN-SR04T seperti pada Gambar 5.26. Gambar 5.26 terdapat pcb untuk menghubungkan antara arduino nano dengan sensor kedalaman air. Hal tersebut karena menurut peneliti bertujuan untuk mempermudah dan memperingskas proses menghubungkan antara arduino nano dengan sensor kedalaman air.



Gambar 5.26 Implementasi Skema Sensor JSN-SR04T



## 5. Implementasi Skema Sensor Kekeruhan *Turbidity*

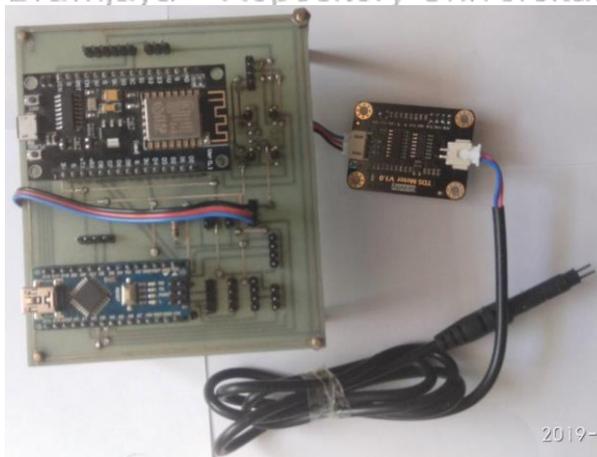
Pada implementasi skema sensor *turbidity* dibuat sesuai pada bab perancangan di Gambar 5.2. implementasi *turbidity* dihubungkan dengan arduino nano yang ditujukan agar nilai dari pembacaan sensor *turbidity* dapat dibaca oleh arduino nano sebagai *microcontroller* berupa nilai *voltage* yang akan dikonversi menjadi nilai NTU dengan rumus regresi linier. Sensor *turbidity* dihubungkan dengan arduino menggunakan kabel jumper dengan pin Vin, *ground* dan pin Analog A3. Berikut foto dari implementasi skema sensor *turbidity* seperti pada Gambar 5.27.



Gambar 5.27 Implementasi Skema Sensor *Turbidity*

## 6. Implementasi Skema Sensor TDS

Pada implementasi skema sensor TDS dibuat sesuai pada bab perancangan di Gambar 5.2. implementasi sensor TDS dihubungkan dengan arduino nano yang ditujukan agar nilai dari pembacaan sensor TDS dapat dibaca oleh arduino nano sebagai *microcontroller* berupa nilai PPM hasil nilai akuisisi dari sensor. Sensor TDS dihubungkan dengan arduino menggunakan kabel jumper dengan pin Vin, *ground* dan pin Analog A2. Berikut foto dari implementasi skema sensor TDS seperti pada Gambar 5.28.



Gambar 5.28 Implementasi Skema Sensor TDS



### 5.2.3 Implementasi Perangkat Lunak

Implementasi perangkat lunak merupakan tahapan implementasi kode program pada sistem agar berjalan sesuai dengan perancangan yang telah dibuat. Tahap implementasi perangkat lunak akan dibagi menjadi beberapa bagian berdasarkan fungsi atau perangkat keras yang diprogram.

#### 1. Implementasi Keseluruhan Sistem

Implementasi keseluruhan sistem adalah sebuah program untuk menjalankan sistem secara keseluruhan. Sistem yang dibuat memiliki 2 program utama yaitu program pada arduino nano untuk melakukan proses akuisisi data dari 5 sensor yang digunakan. Selain itu terdapat program untuk node MCU untuk melakukan klasifikasi metode k-NN dan melakukan proses mengirim data ke firebase serta melakukan proses mengambil data dari firebase untuk menggerakkan purwarupa alat mengelilingi bendungan air.

#### 2. Implementasi Fungsi Deteksi Sensor pH-014

Implementasi fungsi deteksi sensor pH-014 merupakan program untuk membaca akuisisi data sensor pH-014. Sensor pH digunakan untuk mendeteksi nilai pH dalam air dan nilai pH air menjadi salah satu elemen penting dalam klasifikasi k-NN dalam menentukan kategori air yang di uji. Baris program untuk melakukan akuisisi data sensor pH seperti Tabel 5.13.

**Tabel 5.13 Baris Program Akuisisi Sensor pH**

No	Baris Program
1	<code>float dataAdc, dataTegangan, datapH;</code>
2	
3	<code>void pH Air() {</code>
4	<code>for(int i=0; i&lt;800; i++) {</code>
5	<code>dataAdc = analogRead(A1);</code>
6	<code>dataTegangan += dataAdc * (5.0 / 1023.0);</code>
7	<code>}</code>
8	
9	<code>dataTegangan = dataTegangan/800;</code>
10	<code>datapH = (3.250 * dataTegangan) - 3.636;</code>
11	<code>Data2[0] = datapH;</code>
12	<code>}</code>

Penjelasan baris program di atas seperti pada Tabel 5.13 adalah

1. Baris 1 : inisialisasi variabel `dataAdc`, `dataTegangan` dan `datapH` dengan jenis `float` yang digunakan untuk proses selanjutnya. `dataAdc` digunakan untuk menyimpan data dari pembacaan pin Analog A1 sensor pH. `dataTegangan` digunakan untuk merubah nilai analog pembacaan sensor menjadi voltase dan `dataTegangan` pada baris 10 berfungsi untuk mencari rata-rata hasil pembacaan sensor pH.
2. Baris 3-7 : perulangan `for` untuk mendapatkan nilai dari sensor pH melalui `analogRead` pin A1 sebanyak 800 kali. Perulangan sebanyak 800 kali digunakan karena nilai dari sensor pH banyak terdapat *noise* sehingga diperlukan banyak pembacaan sensor.



3. Baris 9 : nilai rata-rata dari tegangan yang didapatkan dari sensor dengan membaginya 800.
4. Baris 10 : penentuan nilai pH air dengan memasukkan nilai variabel tegangan menuju nilai regresi linier. Rumus regresi linier yang didapatkan adalah  $\text{data}_{\text{pH}} = (-3.250 * \text{data}_{\text{Tegangan}}) + 3.636$ ; . Nilai pada rumus regresi linier ini didapatkan dengan melakukan kalibrasi sensor dengan air pH 4, 7, 9 sebelum melakukan pengujian sensor pH.
5. Baris 10 : proses menyimpan  $\text{data}_{\text{pH}}$  yang didapatkan menuju array  $\text{Data2}$  dengan *index* ke 0.  $\text{Array Data}[0]$  akan digunakan dalam proses klasifikasi k-NN pada node MCU.

### 3. Implementasi Fungsi Deteksi Sensor Suhu DS18B20

Implementasi fungsi deteksi sensor suhu DS18B20 merupakan sebuah program untuk melakukan akuisisi data sensor suhu. Suhu air merupakan salah satu element penting dalam menentukan klasifikasi kualitas air. Baris program yang digunakan dalam akuisisi sensor suhu seperti pada Tabel 5.14.

Tabel 5.14 Baris Program Akuisisi Sensor Suhu

No	Baris Program
1	#include <OneWire.h>
2	#include <DallasTemperature.h>
3	OneWire data(4);
4	DallasTemperature datasuhu(&data);
5	
6	void setbegin(){
7	sensors.begin();
8	}
9	
10	void Suhu_Air(void){
11	data.requestTemperatures();
12	float dataSUHU = datasuhu.getTempCByIndex(0);
13	Data2[1] = dataSUHU;
14	}

Penjelasan baris program akuisisi sensor suhu seperti pada Tabel 5.14 adalah

1. Baris 1-2 : inialisasi *library* yang digunakan untuk akuisisi data sensor suhu.
2. Baris 4-5 : proses inialisasi data *library* yang digunakan.
3. Baris 12-13 : proses akuisisi data sensor suhu menggunakan *library* yang digunakan. Data hasil akuisisi sensor suhu akan disimpan pada variabel  $\text{dataSUHU}$ .
4. Variabel  $\text{dataSUHU}$  akan disimpan pada *array*  $\text{Data2}$  dengan *index* ke 1 yang selanjutnya akan digunakan untuk klasifikasi k-NN.

### 4. Implementasi Fungsi Deteksi Sensor Kedalaman JSN-SR04T

Implementasi fungsi deteksi sensor kedalaman JSN-SR04T merupakan sebuah program untuk melakukan akuisisi data sensor kedalaman JSN-SR04T. Sensor kedalaman memiliki fungsi untuk mengetahui kedalaman dari bendungan air dan kedalaman air tersebut akan digunakan sebagai salah satu variabel untuk proses

klasifikasi k-NN. Baris program dalam akuisisi data sensor kedalaman bisa dilihat seperti pada Tabel 5.15.

**Tabel 5.15 Baris Program Akuisisi Sensor Kedalaman**

No	Baris Program
1	#define trigger 7
2	#define echo 8
3	
4	long durasi;
5	int jarak;
6	
7	void setOUTPUT() {
8	pinMode(trigger, OUTPUT);
9	pinMode(echo, INPUT);
10	}
11	void Kedalaman_Air() {
12	digitalWrite(trigger, LOW);
13	delayMicroseconds(5);
14	digitalWrite(trigger, HIGH);
15	delayMicroseconds(10);
16	digitalWrite(trigger, LOW);
17	
18	durasi = pulseIn(echo, HIGH);
19	
20	jarak = durasi*0.034/2;
21	
22	Data2[2] = jarak * 4.4;
23	}

Penjelasan baris program akuisisi sensor kedalaman seperti pada Tabel 5.15 adalah

1. Baris 1-2 : inialisasi pin yang akan digunakan untuk sensor kedalaman. Pin digital 7 untuk pin trigger dan pin digital 8 untuk pin echo sensor.
2. Baris 4-5 : inialisasi variabel yang akan digunakan untuk proses akuisisi sensor. Variabel durasi dengan *type* data long dan variabel jarak dengan *type* data integer.
3. Baris 12 – 18 : proses akuisisi sensor kedalaman. Proses akuisisi dimulai dengan melakukan set *trigger* LOW selama 5 microsecond lalu di set HIGH selama 10 microsecond dan selanjutnya pin *trigger* diset LOW. Variabel durasi digunakan untuk menyimpan data seberapa lama pin echo memancarkan sinar ultrasonik hingga diterima pin *trigger*.
4. Baris 20 : perhitungan konversi waktu pada variabel durasi menjadi jarak centimeter.
5. Baris 22 : proses menyimpan data dari sensor kedalaman ke *array* Data dengan *index* ke 2 yang akan digunakan ke dalam proses selanjutnya yaitu proses klasifikasi k-NN. Nilai jarak di kali 4,4 karena kecepatan rambat ultrasonik 4,4 lebih cepat dari pada di atas air.

##### 5. Implementasi Fungsi Deteksi Sensor Kekeruhan *Turbidity*

Implementasi fungsi deteksi sensor kekeruhan *turbidity* merupakan sebuah program untuk melakukan akuisisi data sensor kekeruhan. Kekeruhan air merupakan salah satu faktor yang penting dalam proses menentukan kualitas air.



Sensor kekeruhan memiliki kerja dengan 2 bagian pada sensor, pada bagian sisi yang lain bertugas sebagai *transmitter* dan sisi yang lain sebagai *receiver*. Proses menentukan kekeruhan air dengan menghitung *output* yang diterima, semakin keruh air akan membuat semakin sedikit data yang diterima. Baris program untuk akuisisi data sensor kekeruhan seperti pada Tabel 5.16.

**Tabel 5.16 Baris Program Akuisisi Sensor Kekeruhan**

No	Baris Program
1	<code>float ntu, volt, NTU;</code>
2	<code>void Kekeruhan Air(){</code>
3	<code>  volt = 0;</code>
4	<code>  for(int i=0; i&lt;800; i++){</code>
5	<code>    int sensorvalue = analogRead(A3);</code>
6	<code>    volt += sensorvalue * (5.0 / 1023.0);</code>
7	<code>  }</code>
8	<code>  volt = volt/800;</code>
9	<code>  ntu = (- 65.20 * volt) + 280;</code>
10	<code>  NTU = floor(ntu);</code>
11	<code>  Data2[3] = NTU;</code>
12	<code>}</code>

Penjelasan baris program untuk akuisisi data sensor kekeruhan seperti Tabel 5.16 adalah

1. Baris 1 : inisialisasi variabel *ntu* dan *volt* dengan *type* data *float*. Variabel *volt* digunakan untuk menyimpan data hasil voltase dari sensor. Variabel *ntu* digunakan untuk menyimpan data hasil konversi nilai voltase menjadi *ntu*.
2. Baris 4-7 : melakukan proses akuisisi data sensor kekeruhan dengan perulangan sebanyak 800 kali. Hal tersebut karena sensor kekeruhan terdapat banyak *noise* pada akuisisi data sensor nya.
3. Baris 9 : proses konversi data voltase menjadi data NTU dengan rumus regresi linier. Rumus regresi linier nya adalah  $ntu = (-65.20 * volt) + 280$ . Angka yang ada pada rumus regresi linier didapatkan dari proses kalibrasi sebelumnya. Proses kalibrasi dengan melakukan melihat nilai pembacaan *output* sensor ketika di beri ke air keruh dengan tingkatan yang berbeda-beda.
4. Baris 10 : proses floor merubah data float *ntu* dengan 2 angka dibelakang koma menjadi data float NTU tidak ada angka dibelakang koma.
5. Baris 11 : proses menyimpan data sensor kekeruhan ke *array* *Data2[3]* dengan *index* ke 3. Data *array* tersebut akan digunakan untuk klasifikasi k-NN yang dilakukan oleh node MCU.

#### 6. Implementasi Fungsi Deteksi Sensor TDS

Implementasi fungsi deteksi sensor TDS merupakan sebuah program untuk melakukan akuisisi data TDS yang memiliki *output* PPM. TDS merupakan salah satu variabel yang penting dalam menentukan klasifikasi kualitas air. Baris program untuk melakukan proses akuisisi sensor TDS bisa dilihat seperti Tabel 5.17.

**Tabel 5.17 Baris Program Akuisisi Sensor TDS**

No	Baris Program
----	---------------



```

1 #include <EEPROM.h>
2 #include "GravityTDS.h"
3
4 GravityTDS tds;
5 float suhu = 25, nilai = 0;
6
7 void TDSbegin(){
8     tds.setPin(A2);
9     tds.setAref(5.0);
10    tds.setAdcRange(1024);
11    tds.begin();
12 }
13
14 void TDS_Air()
15 {
16     tds.setTemperature(suhu);
17     nilai = tds.getTdsValue();
18     Data2[4] = nilai;
19 }

```

Penjelasan dari setiap baris program dalam akuisisi data pada sensor TDS seperti di bawah ini:

1. Baris 1-2 : inialisasi *library* yang akan digunakan untuk melakukan akuisisi sensor TDS yaitu EEPROM.h dan GravityTDS.h. EEPROM.h digunakan sebagai penyimpanan data ketika melakukan kalibrasi sensor sebelum pengujian. GravityTDS.h merupakan *library* yang sudah disediakan oleh pabrik pembuat untuk mempermudah pengguna dalam mengoperasikan sensor TDS.
2. Baris 4 : inialisasi variabel tds untuk mewakili *library* GravityTDS.
3. Baris 5 : inialisasi variabel suhu = 25 dan dengan *type* float serta variabel nilai = 0. Variabel suhu digunakan untuk variabel referensi suhu yang digunakan di dalam *library* dan variabel nilai digunakan untuk menyimpan hasil proses akuisisi data sensor TDS.
4. Baris 9-12 : inialisasi *library* sebelum mengakuisisi sensor. Tds.SetPin digunakan untuk mengetahui pin mana sensor TDS digunakan. Tds.setAref merupakan analog referensi di mana sensor TDS menggunakan 5V sebagai analog referensinya. Tds.setAdcRange digunakan sebagai menentukan *range* ADC dari sensor sebesar 1024, *range* tersebut digunakan karena arduino nano memiliki resolusi 10 bit ADC *range*.
5. Baris 16-17 : proses mengakuisisi data pada sensor TDS dengan memanggil *library* dengan *syntax* tds.setTemperature(suhu) dan tds.getTdsValue().
6. Baris 18 : Hasil akuisisi data sensor disimpan di variabel nilai dan akan dimasukkan ke dalam *array* Data2[4] dengan *index* ke 4. Data *array* tersebut akan digunakan untuk proses klasifikasi k-NN.
7. Implementasi Pengiriman Data Arduino ke Node MCU

Program pengiriman data dari arduino nano ke node MCU merupakan proses pengiriman data 5 sensor hasil akuisisi data pada arduino nano ke node MCU untuk



diproses klasifikasi k-NN dan diproses kirim ke firebase. Baris program nya seperti pada Tabel 5.18 di bawah ini.

**Tabel 5.18 Baris Program Mengirim Data Arduino Nano ke Node MCU**

No	Baris Program
1	String payload = "{}";
2	payload += "\"ph\":";
3	payload += Data2[0];
4	payload += "\",\"suhu\":";
5	payload += Data2[1];
6	payload += "\",\"kedalaman\":";
7	payload += Data2[2];
8	payload += "\",\"kekeruhan\":";
9	payload += Data2[3];
10	payload += "\",\"TDS\":";
11	payload += Data2[4];
12	payload += "\",\"counter\":";
13	payload += conter;
14	payload += "};
15	char attribute [200];
16	payload.toCharArray( attribute, 200 );
17	Serial.println(attribute);
18	
19	

Baris Program pengiriman data arduino ke node MCU penjelasannya seperti berikut ini:

1. Baris 2 – 15 : proses menyimpan data 5 sensor dan *counter* menjadi sebuah file payload dengan *type* data string.
2. Baris 16 : inialisasi variabel *attribute* dengan *type* char sebesar 200.
3. Baris 17 : proses mengubah data ke char *array* dengan isi data variabel *attribute* sebesar 200.
4. Baris 18 : proses mengirim data dari arduino nano ke node MCU.

**8. Implementasi Proses Menerima Data Node MCU dari Arduino Nano**

Proses menerima data pada node MCU adalah melakukan proses memecah data yang kirim dari arduino nano dari 1 paket data menjadi beberapa paket. Paket yang dipecah akan di masukkan ke data 5 sensor yang akan digunakan node MCU sebagai data klasifikasi k-NN dan mengirim data 5 sensor tersebut ke firebase. Baris program dapat dilihat seperti Tabel 5.19 di bawah ini.

**Tabel 5.19 Baris Program Menerima Data Node MCU dari Arduino Nano**

No	Baris Program
1	void recvWithEndMarker() {
2	static byte indx = 0;
3	char endMarker = '\n';
4	char zx;
5	
6	while (Serial.available() > 0 && newData == false) {
7	



```

8   zx = Serial.read();
9   if (zx != endMarker) {
10  receivedChars[indx] = zx;
11  ndx++;
12  if (indx >= numChars) {
13  indx = numChars - 1;
14  }
15  } else {
16  receivedChars[indx] = '\0';
17  indx = 0;
18  newData = true;
19  }
20 }
21 }
22
23 void LihatData() {
24   if (newData == true) {
25     Serial.println(receivedChars);
26     input = receivedChars;
27     newData = false;
28   }
29 }
30

```

Penjelasan baris program untuk menerima data node MCU sebagai berikut ini.

1. Baris 2 - 5 adalah proses inialisasi void dan variabel yang digunakan untuk proses selanjutnya
2. Baris 7- 20 merupakan proses memecah data dari 1 paket yang dikirim arduino nano menjadi data untuk inialisasi sensor.
3. Baris 23 – 28 melakukan proses menampilkan data yang sudah diterima dan di pecah oleh node MCU untuk menjadi data di proses selanjutnya.

### 9. Implementasi Klasifikasi k-NN

Implementasi proses klasifikasi k-NN memiliki beberapa tahapan dimulai dengan penentuan nilai K hingga hasil akhir klasifikasi sesuai pada Gambar 5.9. Baris program untuk proses klasifikasi k-NN seperti pada Tabel 5.20 di bawah ini.

**Tabel 5.20 Baris Program Klasifikasi k-NN**

No	Baris Program
1	
2	float Data[5];
3	float jarak[30][2];
4	float DataNormalisasi[5];
5	int DataNormalisasiINT[5];
6	float DataNormalisasiFinal[5];
7	
8	void loop() {
9	start = millis();
10	
11	//Normalisasi min-max float
12	DataNormalisasi[0] = (Data2[0]-0)/14; //0-14
13	DataNormalisasi[1] = (Data2[1]-0)/85; //0-100
14	DataNormalisasi[2] = (Data2[2]-20)/450; //20-1011
15	DataNormalisasi[3] = (Data2[3]-0)/1000; //0-3000
16	DataNormalisasi[4] = (Data2[4]-0)/1000; //0-1000



```

17
18 //Normalisasi float to int dan int to float
19 DataNormalisasiINT[0] = round(DataNormalisasi[0] * 1000);
20 DataNormalisasiFinal[0] = (float)DataNormalisasiINT[0] /
21 1000;
22 DataNormalisasiINT[1] = round(DataNormalisasi[1] * 1000);
23 DataNormalisasiFinal[1] = (float)DataNormalisasiINT[1] /
24 1000;
25 DataNormalisasiINT[2] = round(DataNormalisasi[2] * 1000);
26 DataNormalisasiFinal[2] = (float)DataNormalisasiINT[2] /
27 1000;
28 DataNormalisasiINT[3] = round(DataNormalisasi[3] * 1000);
29 DataNormalisasiFinal[3] = (float)DataNormalisasiINT[3] /
30 1000;
31 DataNormalisasiINT[4] = round(DataNormalisasi[4] * 1000);
32 DataNormalisasiFinal[4] = (float)DataNormalisasiINT[4] /
33 1000;
34
35 Serial.println("Hasil Perhitungan Euclidean");
36 int DataJarakHitung = 0;
37 int label = 1;
38 for(int a=0;a<50;a++){
39 jarak[a][0] = 0; //reset data
40 for(int b=0;b<5;b++){
41 jarak[a][0] += pow( DataNormalisasiFinal[b] -
42 Datalatih(a,b),2); //data uji - data latih(ph, kedalaman,
43 kejernihan, suhu, tds)
44 }
45 jarak[a][0] = sqrt(jarak[a][0]); //akar
46 switch(a){
47 case 0 ... 9:
48 jarak[a][1] = 0; //angka 0 untuk label jelek
49 break;
50 case 10 ... 29:
51 jarak[a][1] = 1; //angka 1 untuk label sedang
52 break;
53 case 30 ... 49:
54 jarak[a][1] = 2; //angka 2 untuk label bersih
55 break;
56 }
57 }
58 Serial.println("-----");
59 Serial.println("Sorting"); //SORTING DATA
60 for(int i=0;i<50;i++){
61 for(int j=i+1;j<50;j++){
62
63 if(jarak[i][DataJarakHitung]>jarak[j][DataJarakHitung]){
64 float temp = jarak[i][DataJarakHitung];
65 jarak[i][DataJarakHitung] = jarak[j][DataJarakHitung];
66 jarak[j][DataJarakHitung] = temp;
67 jarak[j][label] = jarak[i][label];
68 jarak[i][label] = jarak[j][label];
69 temp = jarak[i][label];
70 jarak[i][label] = jarak[j][label];
71 jarak[j][label] = temp;
72 }
73 }
74 }
75

```



```

76 float satu = 0, dua = 0, tiga = 0;
77 String hasil;
78 //Serial.println("K = 5");
79 for(int a=0;a<5;a++){
80 Serial.println("Perhitungan label");
81 if(jarak[a][label] == 0.00)
82 {
83 satu += 1/ pow(jarak[a][DataJarakHitung],2);
84 }
85 else if(jarak[a][label] == 1.00)
86 {
87 dua += 1/ pow(jarak[a][DataJarakHitung],2);
88 }
89 else if(jarak[a][label] == 2.00)
90 {
91 tiga += 1/ pow(jarak[a][DataJarakHitung],2);
92 }
93 }
94
95 if(satu > dua && satu > tiga){
96 hasil = "JELEK";
97 }
98 else if(dua > satu && dua > tiga){
99 hasil = "SEDANG";
100 }
101 else if(tiga > dua && tiga > satu){
102 hasil = "BAGUS";
103 }
104 Serial.println("-----");
105 Serial.print("HASIL :");
106 Serial.println(hasil);
107 finish = millis();
108 Serial.print("WAKTU :");
109 waktu = (finish - start)/1000.0;
110 Serial.println(waktu);
111 .....

```

Baris program proses klasifikasi k-NN seperti pada Tabel 5.20 di atas ini memiliki proses secara bertahap. Adapun penjelasan baris program dari berbagai tahapannya seperti di bawah ini.

1. Baris 2-6 : inisialisasi variabel `Data2[5]`, `jarak[5]`, `DataNormalisasi[5]`, `DataNormalisasiFinal[5]`. Array `Data2[5]` yang digunakan untuk menyimpan hasil akuisisi lima sensor yang digunakan dengan *type* data float. Variabel `jarak[50][2]` digunakan untuk menyimpan data hasil perhitungan *euclidean* jarak data uji dengan data latih. `DataNormalisasi[5]` untuk menyimpan data hasil normalisasi, `DataNormalisasiFinal[5]` untuk data normalisasi final yang akan digunakan untuk klasifikasi k-NN.
2. Baris 9: inisialisasi `millis` untuk memulai waktu komputasi dari proses klasifikasi k-NN sehingga bisa diketahui seberapa lama waktu satu proses klasifikasi k-NN hingga selesai.
3. Baris 19 – 33 : proses normalisasi data 5 sensor yang diterima node MCU dari arduino nano. Proses normalisasi diperlukan pada klasifikasi k-NN karena *range* data tiap sensor berbeda-beda sehingga perlu dinormalisasi terlebih dahulu.



4. Baris 37 – 56 : proses perhitungan *euclidean* yang digunakan pada klasifikasi k-NN untuk mengetahui jarak data uji dengan data latih yang telah disediakan dan proses untuk memberikan label hasil data proses *euclidean*.
5. Baris 58 – 74 : proses sorting *Ascending* data hasil perhitungan *euclidean*.
6. Baris 79 – 103 : proses mengambil data sejumlah K yang paling kecil dengan nilai K telah ditentukan sebelumnya dan melakukan proses *voting weighted* pada proses klasifikasi sehingga didapat hasil yang optimal.
7. Baris 104 - 111 : menampilkan nilai variabel hasil dan variabel waktu akan di stop untuk mengetahui seberapa lama sistem memproses klasifikasi k-NN.

**10. Implementasi Pengiriman Data Node MCU ke Firebase**

Implementasi pengiriman data node MCU merupakan implementasi program untuk proses mengirim data yang diterima node MCU dari arduino nano menuju firebase. Program tersebut seperti pada Tabel 5.21.

**Tabel 5.21 Baris Program Mengirim Data Node MCU**

No	Baris Program
1	
2	#include <ESP8266WiFi.h>
3	#include <ArduinoJson.h>
4	#include <FirebaseArduino.h>
5	
6	#define NAMA_WIFI "WIFI GRATIS"
7	#define PASS_WIFI "123456789"
8	#define ALAMAT_FIREBASE "skripsi-e3d0d.firebaseio.com"
9	#define PASS_FIREBASE "4XD8gHxqNZDP3Cluqbkwgsrftasdac"
10	
11	void setup() {
12	Serial.begin(9600);
13	WiFi.begin(NAMA_WIFI, PASS_WIFI);
14	Serial.println(WiFi.localIP());
15	
16	Firebase.begin(NAMA_FIREBASE, PASS_FIREBASE);
17	}
18	
19	void loop() {
20	
21	Firebase.setString("PHAIR", pHAir);
22	Firebase.setString("SuhuAIR", SuhuAir);
23	Firebase.setString("KedalamanAIR", kedalamanAir);
24	Firebase.setString("KekeruhanAIR", kekeruhanAir);
25	Firebase.setString("TDSAIR", TDSAir);
26	Firebase.setString("Hasil Monitoring", Hasil);
27	
28	}

Baris Program pada node MCU untuk proses mengirim data menuju firebase dapat dijelaskan seperti berikut ini.

1. Baris 2 – 5 : inisialisasi *Library* #include <ESP8266WiFi.h> #include <PubSubClient.h> #include <ArduinoJson.h> #include <FirebaseArduino.h>. *esp8266WiFi.h* digunakan untuk *library* menjalankan modul *esp8266*. *pubSubClient.h* digunakan untuk *library* menerima data dari arduino nano. *Library arduinojson.h* digunakan untuk



proses menerima data dari arduino yang melalui RX TX dan metode UART. FirebaseArduino.h merupakan *library* untuk proses mengirim data menuju firebase.

2. Baris 6 - 9 : mendefinisikan firebase yang dituju *host* dengan nama firebasenya adalah skripsi-e3d0d.firebaseio.com dan firebase *auth* nya adalah 4XD8gHxqNZDP3C1uqbkwgsrftasdac. Hal tersebut sangat penting untuk mengetahui alamat firebase yang dituju..
3. Baris 13 – 16 : inialisasi untuk *library* agar node MCU bisa terkoneksi dengan WiFi dan firebase yang sudah di inialisasi pada baris 6-9.
4. Baris 21 – 26 : proses pengiriman data sensor dan data klasifikasi menuju firebase. Struktur data yang dikirim ke firebase adalah data pHAir akan dikirim ke firebase bagian “PHAIR”, data SuhuAir akan dikirim ke firebase bagian “SuhuAIR”, data kedalamanAir akan dikirim ke firebase bagian “KedalamanAIR”, data kekeruhan Air akan dikirim ke firebase bagian “KekeruhanAIR”, data TDSAir akan dikirim ke firebase bagian “TDSAIR”, data hasil akan dikirim ke firebase bagian “Hasil Monitoring”.

**11. Implementasi Proses Node MCU Menerima Data dari Aplikasi**

Proses menerima data dari firebase yang dilakukan oleh node MCU dimulai dari mengambil data terlebih dahulu. Setelah data diterima oleh node MCU maka data tersebut akan diproses sebagai perintah untuk menggerakkan motor DC sebagai penggerak purwarupa alat. Baris program untuk proses menerima data untuk pergerakan purwarupa alat seperti pada Tabel 5.22 di bawah ini.

**Tabel 5.22 Baris Program Untuk Pergerakan Purwarupa Alat**

No	Baris Program
1	
2	String Gerak1 = Firebase.getString("Gerak");
3	Serial.println(Gerak1);
4	

Penjelasan dari baris program di atas adalah

1. Baris 2 adalah proses node MCU mengambil data firebase “Gerak” dan menyimpan kedalam sebuah variabel yang bernama Gerak1.
2. Baris 3 menampilkan data dari firebase tersebut.

**12. Implementasi Proses Pergerakan Purwarupa Alat**

Pada implementasi proses pergerakan purwarupa alat merupakan proses menerjemahkan data dari firebase yang dilakukan oleh node MCU untuk menjadi pergerakan motor DC. Baris program proses node MCU mengatur pergerakan motor DC sesuai perintah dapat dilihat seperti Tabel 5.23.

**Tabel 5.23 Baris Program Proses Pergerakan Purwarupa Alat**

No	Baris Program
1	
2	if(Gerak1 == "0"){ //STOP
3	digitalWrite(MOTOR1A, LOW);



```

4 digitalWrite (MOTOR1B, LOW);
5 digitalWrite (MOTOR2A, LOW);
6 digitalWrite (MOTOR2B, LOW);
7 Serial.println ("STOP");
8 }else if (Gerak1 == "1") { //MAJU
9 digitalWrite (MOTOR1A, LOW);
10 digitalWrite (MOTOR1B, HIGH);
11 digitalWrite (MOTOR2A, LOW);
12 digitalWrite (MOTOR2B, HIGH);
13 Serial.println ("MAJU");
14 }else if (Gerak1 == "2") { //MUNDUR
15 digitalWrite (MOTOR1A, HIGH);
16 digitalWrite (MOTOR1B, LOW);
17 digitalWrite (MOTOR2A, HIGH);
18 digitalWrite (MOTOR2B, LOW);
19 Serial.println ("MUNDUR");
20 }else if (Gerak1 == "3") { //KIRI
21 digitalWrite (MOTOR1A, LOW);
22 digitalWrite (MOTOR1B, HIGH);
23 digitalWrite (MOTOR2A, HIGH);
24 digitalWrite (MOTOR2B, LOW);
25 Serial.println ("KIRI");
26 }else if (Gerak1 == "4") { //KANAN
27 digitalWrite (MOTOR1A, HIGH);
28 digitalWrite (MOTOR1B, LOW);
29 digitalWrite (MOTOR2A, LOW);
30 digitalWrite (MOTOR2B, HIGH);
31 Serial.println ("KANAN");
32 }
33 }

```

Penjelasan baris program di atas sebagai berikut :

1. Baris 2 – 7 : proses menggerakkan purwarupa alat dengan keadaan STOP jika node MCU menerima data bernilai "0".
2. Baris 8 – 13 : proses menggerakkan motor DC agar purwarupa alat bergerak MAJU jika node MCU mendapatkan data dari firebase bernilai "1".
3. Baris 14 – 19 : proses menggerakkan purwarupa alat dengan pergerakan MUNDUR bila aplikasi *smartphone* mengirim data bernilai "2".
4. Baris 20 – 25 : proses menggerakkan purwarupa alat dengan arah ke KIRI jika node MCU mendapatkan data dari firebase bernilai "3".
5. Baris 26 – 32 : proses jika node MCU menerima data bernilai "4" maka node MCU akan menggerakkan motor DC supaya purwarupa alat dapat bergerak ke KANAN.

### 13. Implementasi Aplikasi *Smartphone*

Desain aplikasi yang dibuat berupa halaman awal seperti pada Gambar 5.28 di atas. Halaman awal akan muncul pertama kali ketika aplikasi dibuka oleh pengguna dengan terdapat 4 fitur. Fitur yang dibuat difungsikan agar mempermudah pengguna dalam mengoperasikan purwarupa alat dalam *monitoring* kualitas air. Fitur pada halaman awal terdiri dari Fitur Informasi Kualitas Air, Fitur *Monitoring* Kualitas Air, Fitur *Histori Monitoring* Air.



Gambar 5.29 Implementasi Desain Aplikasi Halaman Awal Smartphone

**Informasi Tentang Parameter Kualitas AIR**

Parameter : pH Air

pH adalah kepanjangan dari potensial hidrogen. pH sendiri merupakan suatu ukuran konsentrasi dari ion hydrogen disebuah larutan air. Ph sendiri salah satu parameter faktor lingkungan yang bisa mempengaruhi pertumbuhan dan kehidupan didalam air. Air murni bersifat netral dengan kadar pH 7. Air yang memiliki larutan pH kurang dari 7 bisa disebut air yang bersifat asam dan ketika air yang memiliki larutan ph lebih dari 7 bisa disebut air yang bersifat basa atau alkali. Kondisi air yang baik menurut peraturan pemerintahan RI no 82 tahun 2001 memiliki rentang nilai 6 hingga 9.

**MONITORING KUALITAS AIR**

7.5  
pH

25.42 °C  
SUHU

210  
PPM

220 CM  
KEDALAMAN AIR

0 NTU  
KEKERUHAN AIR

**HASIL : Bagus**

MAJU

KIRI

STOP

KANAN

MUNDUR

79

B

SIMPAN

**HASIL PENGUKURAN KUALITAS AIR**

Tanggal	PH	Suhu	Kedalaman	Kekeruhan	Tds
2020-01-25 08:43:47	7.73	25.50 °C	221.00 CM	10.00 NTU	165.99
2020-01-25 08:49:46	7.73	25.50 °C	288.00 CM	0.3 NTU	185.99
2020-01-25 09:07:16	7.73	25.70 °C	252.20 CM	10.00 NTU	265.99
2020-01-25 10:15:57	8.73	27.29 °C	132.00 CM	50.00 NTU	330.99
2020-01-25 09:24:57	8.30	24.30 °C	222.00 CM	100.00 NTU	360.00
2020-01-25 09:27:27	7.99	25.39 °C	278.00 CM	50.00 NTU	299.00
2020-01-25 09:31:13	8.05	23.21 °C	242.00 CM	100.00 NTU	315.00
2020-01-25 09:40:46	7.62	25.08 °C	299.20 CM	10.00 NTU	176.00
2020-01-25 09:44:09	8.61	24.97 °C	281.60 CM	50.00 NTU	350.00

Gambar 5.30 Implementasi Tampilan isi Aplikasi

Implementasi tampilan isi aplikasi seperti pada Gambar 5.30 A merupakan tampilan yang berisi informasi cara mengetahui kualitas air yang baik menurut peraturan pemerintah. Isi tampilan tersebut berisi informasi pH, Kekeruhan, Kedalaman, Suhu, dan TDS. Tampilan tersebut akan tampilan ketika pengguna meng-klik *bottom* informasi kualitas air pada halaman awal aplikasi.

Gambar 5.30 B berisi parameter yang ditampilkan seperti hasil klasifikasi menggunakan metode k-NN , nilai pH air, nilai suhu air, nilai kekeruhan air, kedalaman air dan jumlah zat terlarut di dalam air. Tampilan tersebut akan tampilan ketika pengguna meng-klik *bottom Monitoring* Kualitas Air pada halaman awal



aplikasi. Tampilan kontrol pergerakan alat yang didesain untuk memudahkan pengguna untuk menggerakkan purwarupa alat tersebut. Purwarupa alat yang dirancang akan bisa bergerak mengelilingi bendungan maka purwarupa alat tersebut harus bisa berjalan maju, mundur, belok kanan, belok kiri dan stop. Selain terdapat tombol yang digunakan untuk menggerakkan purwarupa alat, pada halaman ini terdapat juga informasi dari kelima sensor yang ada di purwarupa alat dan hasil klasifikasi k-NN dari *input* kelima sensor yang digunakan. Tampilan tersebut akan tampilan ketika pengguna meng-klik *bottom* Gerak purwarupa alat pada halaman awal aplikasi. Terdapat tombol simpan yang berfungsi untuk menyimpan data yang ditampilkan pada halaman *history*.

Gambar 5.30 C merupakan tampilan *history* data dari *monitoring* kualitas air yang telah dilakukan sebelumnya. Tampilan ini dibuat agar hasil pengukuran bisa diakses kemudian hari ketika dibutuhkan untuk kegiatan yang lainnya seperti rekap hasil *monitoring* kualitas air. Tampilan tersebut akan tampilan ketika pengguna meng-klik *bottom* Histori *Monitoring* Kualitas Air pada halaman awal aplikasi.

#### 14. Implementasi Data Pada Klasifikasi K-NN

Pada Implementasi data pada klasifikasi k-NN disini berisi inialisasi data latih dan data uji yang digunakan pada penelitian ini. Data latih dan data uji yang digunakan pada penelitian ini dilakukan proses normalisasi min-max sebelum masuk ke proses klasifikasi k-NN. Normalisasi digunakan agar jarak *range* antar data dari 5 sensor yang digunakan tidak berbeda jauh satu sama lain karena data hasil normalisasi akan memiliki *range* 0 hingga 1. Proses implementasi data latih dan data uji untuk klasifikasi k-NN dapat dilihat pada Tabel 5.24.

Tabel 5.24 Data pada Klasifikasi k-NN

No	Baris Program
1	
2	float DataLatih[50][5] = {{0.436,0.296,0.469,0.1,0.410},
3	{0.421,0.317,0.440,0.1,0.530},{0.7,0.363,0.391,0.2,0.498},{
4	0.653,0.347,0.391,0.2,0.450},{0.391,0.317,0.440,0.1,0.624},
5	{0.365,0.349,0.401,0.1,0.581},{0.496,0.353,0.535,0.2,0.561}
6	},{0.419,0.339,0.469,0.1,0.599},{0.441,0.336,0.440,0.2,0.482
7	},{0.626,0.352,0.469,0.5,0.714},
8	{0.439,0.259,0.606,0.05,0.298},{0.484,0.259,0.665,0.05,0.31
9	5},{0.628,0.276,0.636,0.05,0.330},{0.636,0.268,0.626,0.05,0
10	.330},{0.558,0.284,0.489,0.1,0.410},{0.615,0.317,0.557,0.05
11	,0.325},{0.531,0.304,0.684,0.05,0.310},{0.650,0.324,0.557,0
12	.1,0.405},{0.464,0.26,0.636,0.05,0.215},{0.629,0.314,0.636,
13	0.05,0.250},{0.616,0.315,0.655,0.05,0.315},{0.586,0.315,0.6
14	55,0.05,0.299},{0.639,0.322,0.469,0.1,0.310},{0.626,0.319,0
15	.489,0.05,0.325},{0.634,0.325,0.557,0.05,0.340},{0.636,0.32
16	5,0.508,0.05,0.310},{0.445,0.259,0.606,0.1,0.380},{0.564,0.
17	292,0.636,0.05,0.340},{0.610,0.296,0.675,0.1,0.290},{0.604,
18	0.301,0.636,0.1,0.284},
19	{0.534,0.292,0.508,0.001,0.1},{0.556,0.294,0.469,0.001,0.19
20	8},{0.604,0.302,0.557,0.001,0.227},{0.598,0.276,0.665,0.001
21	,0.132},{0.58,0.274,0.489,0.001,0.154},{0.56,0.295,0.655,0.
22	001,0.210},{0.544,0.302,0.469,0.001,0.094},{0.551,0.305,0.4
23	89,0.001,0.190},{0.528,0.292,0.655,0.001,0.212},{0.531,0.28



```

24 4,0.665,0.001,0.184},{0.529,0.288,0.665,0.001,0.179},{0.507
25 ,0.307,0.733,0.001,0.198},{0.591,0.311,0.724,0.01,0.241},{0
26 .558,0.316,0.684,0.001,0.205},{0.585,0.295,0.684,0.001,0.26
27 1},{0.574,0.3,0.606,0.01,0.236},{0.571,0.306,0.469,0.001,0.
28 280},{0.504,0.288,0.557,0.01,0.215},{0.507,0.295,0.469,0.00
29 1,0.123},{0.565,0.301,0.508,0.001,0.251}};
30
31 float DataBagus[10][5] =
32 {{0.604,0.295,0.557,0.0003,0.091},{0.564,0.286,0.469,0.001,
33 0.220},{0.507,0.296,0.491,0.0003,0.180},{0.582,0.306,0.606,
34 0.0003,0.150},{0.536,0.289,0.557,0.0003,0.245},{0.545,0.302
35 ,0.535,0.0003,0.139},{0.559,0.294,0.655,0.0003,0.100},{0.51
36 1,0.292,0.655,0.01,0.101},{0.589,0.271,0.606,0.01,0.198},{0
37 .513,0.276,0.469,0.0003,0.099}};
38
39 float DataSedang[10][5] =
40 {{0.601,0.304,0.606,0.05,0.300},{0.586,0.293,0.655,0.1,0.31
41 5},{0.629,0.308,0.469,0.05,0.340},{0.621,0.296,0.489,0.1,0.
42 354},{0.581,0.299,0.606,0.01,0.311},{0.613,0.308,0.535,0.05
43 ,0.283},{0.565,0.288,0.469,0.05,0.256},{0.560,0.284,0.535,0
44 .05,0.241},{0.640,0.310,0.655,0.1,0.329},{0.613,0.305,0.655
45 ,0.2,0.350}};
46
47 float DataJelek[7][5] =
48 {{0.428,0.311,0.606,0.2,0.430},{0.636,0.331,0.535,0.2,0.398
49 },{0.443,0.324,0.391,0.1,0.453},{0.464,0.340,0.469,0.1,0.51
50 7},{0.5,0.334,0.469,0.2,0.381},{0.439,0.331,0.391,0.1,0.598
51 },{0.492,0.324,0.535,0.1,0.487}};
52

```

Penjelasan *Source Code* dari implementasi data pada klasifikasi k-NN sebagai berikut ini

1. Baris 2 – 7 merupakan data latih yang sudah dinormalisasi dengan label kelas “Jelek” dengan jumlah data 10 data.
2. Baris 8 – 18 merupakan data latih yang sudah dinormalisasi dengan label kelas “Sedang” dengan jumlah data 20 data.
3. Baris 19 – 29 merupakan data latih yang sudah dinormalisasi dengan label kelas “Bagus” dengan jumlah data 20 data.
4. Baris 31 – 37 merupakan data uji yang sudah dinormalisasi dengan label kelas “Bagus” dengan jumlah data 10 data yang akan digunakan untuk menguji akurasi dari klasifikasi k-NN yang digunakan pada penelitian ini.
5. Baris 39 – 45 merupakan data uji yang sudah dinormalisasi dengan label kelas “Sedang” dengan jumlah data 10 data yang akan digunakan untuk menguji akurasi dari klasifikasi k-NN yang digunakan pada penelitian ini.
6. Baris 47 – 51 merupakan data uji yang sudah dinormalisasi dengan label kelas “Jelek” dengan jumlah data 5 data yang akan digunakan untuk menguji akurasi dari klasifikasi k-NN yang digunakan pada penelitian ini.

## BAB 6 PENGUJIAN DAN ANALISIS

Bab pengujian berisi tentang beberapa pengujian sistem yang dibuat beserta alat pendukungnya. Pengujian memiliki tujuan untuk mengetahui sistem yang dibuat sudah memiliki fungsionalitas dan kinerja yang diharapkan atau belum. Pada penelitian ini akan dilakukan pengujian fungsionalitas sistem dan kinerja sistem.

### 6.1 Gambaran Lingkungan Pengujian

Lokasi pengujian yang digunakan dalam penelitian ini berada di 3 lokasi yaitu bendungan air Desa Kresek seperti Gambar 6.1, bendungan di Desa Golang seperti pada Gambar 6.2 dan bendungan di Desa Giringan seperti pada Gambar 6.3. Lokasi ini dipilih peneliti karena fungsi dari bendungan ini sangatlah penting bagi masyarakat sekitar ketika terjadi musim kemarau atau ketika kekurangan air bersih.

Pada pengujian ini dilakukan *monitoring* kualitas air dengan menempatkan alat yang dibuat ke bendungan air. Alat yang dibuat akan melakukan akuisisi data dari kelima sensor secara keseluruhan bendungan air dengan titik titik yang telah ditentukan. Setelah melakukan akuisisi data, alat yang dibuat akan mengirim data akuisisi data tersebut dan data hasil klasifikasi kualitas air tersebut menuju firebase agar data tersebut bisa di akses melalui *smartphone*.



Gambar 6.1 Lokasi Pengujian Alat di Desa Kresek





Untuk mengetahui akurasi sensor pH dengan cara membandingkan nilai pH dari sensor dengan pH tester konvensional sehingga terdapat selisih *error* dari data pada sensor pH yang digunakan. Sebelum pengujian dilakukan proses kalibrasi sensor pH dengan sampel air pH khusus untuk kalibrasi dengan nilai pH air 4, pH air 7, pH air 9. Dengan *output* sensor pH berupa voltase maka *output* sensor dimasukkan dalam rumus regresi linier dengan data hasil kalibrasi. Nilai pada regresi linier sensor pH didapatkan pada saat proses kalibrasi sensor pH. Hasil rumus regresi linier yang didapatkan akan digunakan untuk menjadi rumus mengkonversi *output voltase* menjadi nilai pH dalam pengujian ini.

### 1. Tujuan Pengujian

Tujuan dari pengujian sensor pH yang digunakan sangatlah penting karena dengan melakukan pengujian tersebut kita dapat mengetahui akurasi sensor pH dalam mengambil data secara akurat atau tidak dan dengan pengujian sensor pH dapat diketahui *error* data dari akuisisi sensor yang digunakan dengan membandingkan nilai data sensor dengan nilai data dari pH tester konvensional.

### 2. Prosedur Pengujian

Prosedur pengujian sensor pH yang digunakan sebagai berikut ini:

1. Pengujian sensor pH dilakukan sebanyak sepuluh kali dengan air pH yang bervariasi agar diketahui nilai akurasinya.
2. Hubungkan arduino nano dengan laptop.
3. Menghubungkan sensor pH dengan arduino nano melalui pin *ground*, pin A1 dan pin VCC yang dihubungkan dengan *driver* motor 5V sesuai pada Gambar 5.4 Perancangan Skema Sensor pH-014.
4. Setelah terhubung semua maka jalankan program yang ada pada arduino nano dan nyalakan *driver* motor.
5. Mengamati hasil data akuisisi sensor pH yang ditampilkan di serial monitor arduino IDE selama 10-15 detik karena proses mengakuisisi nilai sensor pH secara bertahap dalam naik maupun turun.
6. Menghitung selisih data sensor dengan alat konvensional dengan Persamaan 6.1

$$\text{Selisih Data} = \text{Data akuisisi sensor} - \text{Data alat konvensional} \quad (6.1)$$

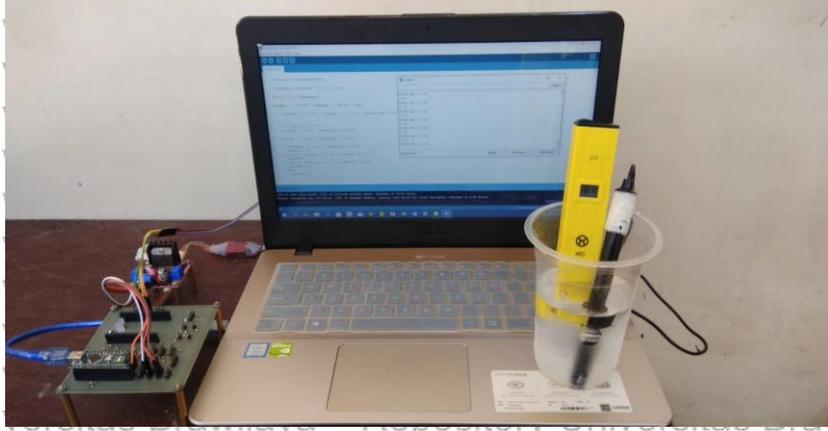
7. Menghitung *error* dari selisih data yang didapat dengan rumus Persamaan 6.2.

$$\text{Error Data} = \frac{\text{Selisih Data}}{\text{Data Alat Konvensional}} \times 100\% \quad (6.2)$$

### 3. Hasil Pengujian dan Analisis

Pada proses pengujian mengakuisisi data sensor pH seperti pada Gambar 6.4 dilakukan dengan cara mencelupkan sensor pH dan pH tester konvensional secara bersamaan di air sebanyak 10 kali. Dengan melakukan pengujian tersebut membuat

peneliti bisa mengetahui sensor yang digunakan memiliki akurasi dengan baik atau tidak. Dengan pengujian sensor yang digunakan dapat menjadi referensi sensor untuk penelitian selanjutnya atau penelitian yang berhubungan dengan pH air. Berikut hasil pengujian sensor pH seperti pada Tabel 6.1 di bawah ini:



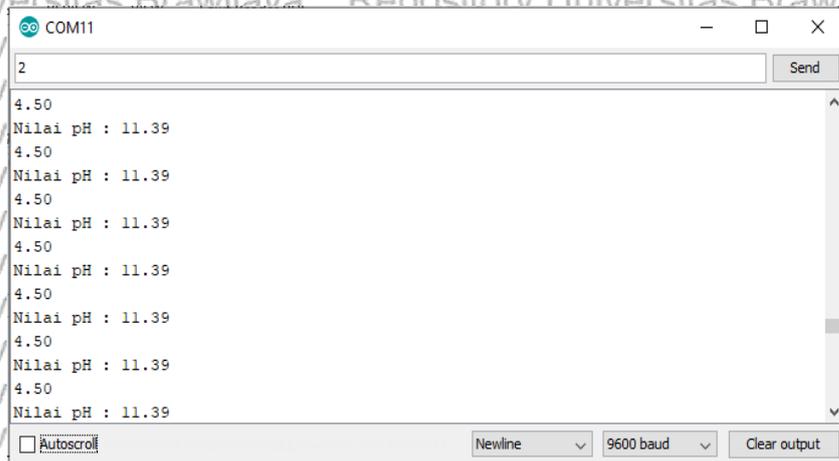
**Gambar 6.4 Pengujian Mengakuisisi Data Sensor pH**

**Tabel 6.1 Tabel Hasil Pengujian Mengakuisisi Nilai Sensor pH**

Pengujian ke	Data pH tester konvensional	Data akuisisi sensor pH-014	Selisih Data	Error Data
1	3,1	3,33	0,23	7,41%
2	4	3,80	0,20	5%
3	4,5	4,88	0,33	7,33%
4	5,4	5,98	0,58	10,74%
5	6,1	6,58	0,48	7,86%
6	7	7,31	0,31	4,42%
7	8,1	8,52	0,42	5,18%
8	9	9,93	0,93	10,33%
9	10,2	11,39	1,19	11,67%
10	11,1	12,19	1,09	9,82%
Rata - rata				7,97%

Hasil pengujian sensor pH yang ditampilkan seperti pada Tabel 6.1 di atas merupakan data dari akuisisi sensor pH yang ditampilkan pada serial monitor. Hasil pengujian sensor yang telah dilakukan didapatkan nilai *error* data sebesar 7,97 %. perbedaan pembacaan sensor pH dengan pH tester memiliki banyak faktor seperti jumlah air yang digunakan untuk kalibrasi. Untuk pH tester sudah dilakukan kalibrasi oleh perusahaan pembuat dan untuk sensor pH dilakukan kalibrasi oleh si peneliti

dengan air pH 3 jenis saja. Hal tersebut kemungkinan membuat terjadi perbedaan pembacaan sensor dengan tester yang cukup besar. Dengan akurasi tersebut menandakan sensor pH yang digunakan pada penelitian ini memiliki akurasi yang baik dan sensor pH tersebut dapat digunakan pada penelitian ini dan selanjutnya. *Output* sensor yang dapat dilihat melalui serial monitor arduino IDE seperti pada Gambar 6.2.



**Gambar 6.5 Hasil Serial Monitor Pengujian Sensor pH-014**

Data pada serial monitor berupa nilai *voltase* dan nilai konversi menjadi nilai Mh dengan rumus persamaan linier. Persamaan linier tersebut didapatkan pada saat dilakukan kalibrasi sensor pH dengan air pH khusus untuk kalibrasi dengan nilai pH air 4, pH air 7 dan pH air 9. Pada Gambar 6.5 merupakan salah satu contoh hasil pengujian sensor pH yang ke-9 pada Tabel 6.1 di atas dengan *output voltase* 4,50 dan nilai pH : 11,39.

### **6.2.2 Pengujian Akuisisi Hasil Data Sensor Suhu**

Pengujian akuisisi hasil data sensor suhu dengan kode TS-02 dilakukan untuk menilai hasil dari sensor suhu yang digunakan pada salah satu data dalam klasifikasi metode k-NN. Proses pengujian sensor suhu menggunakan sampel air dengan suhu air yang bervariasi mulai dari dingin hingga panas agar diketahui akurasi dari sensor suhu yang digunakan. Untuk mengetahui akurasi sensor suhu dengan cara membandingkan nilai suhu dari sensor dengan termometer suhu konvensional sehingga terdapat selisih *error* dari data pada sensor suhu yang digunakan.

#### **1. Tujuan Pengujian**

Tujuan dari pengujian sensor suhu yang digunakan sangatlah penting karena dengan melakukan pengujian tersebut kita dapat mengetahui akurasi sensor suhu dalam mengambil data secara akurat atau tidak dan dengan pengujian sensor suhu dapat diketahui *error* data dari akuisisi sensor yang digunakan dengan membandingkan nilai data sensor dengan nilai data dari termometer suhu konvensional.



## 2. Prosedur Pengujian

Prosedur pengujian sensor suhu sebagai berikut ini :

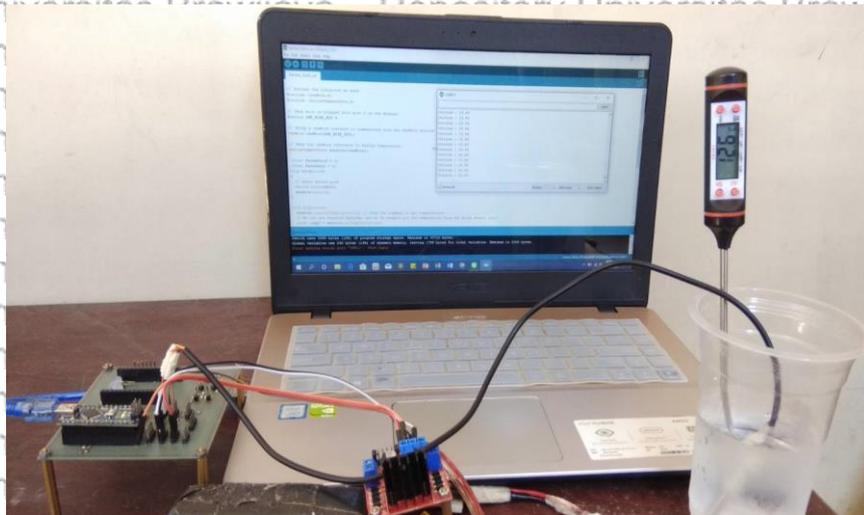
1. Pengujian sensor suhu dilakukan sebanyak sepuluh kali dengan air yang memiliki suhu bervariasi agar diketahui nilai akurasi.
2. Hubungkan arduino nano dengan laptop.
3. Menghubungkan sensor suhu dengan arduino nano melalui pin *ground*, pin Digital D1 dan pin *VCC* yang dihubungkan dengan *driver* motor 5V sesuai pada Gambar 5.5 Perancangan Skema Sensor Suhu DS18B20.
4. Setelah terhubung semua maka jalankan program yang ada pada arduino nano dan nyalakan *driver* motor.
5. Mengamati hasil data akuisisi sensor suhu yang ditampilkan di serial monitor arduino IDE selama 15 detik karena proses mengakuisisi nilai sensor suhu secara bertahap dalam naik maupun turun suhu yang terdeteksi hingga nilai yang diperoleh stabil.
6. Menghitung selisih data sensor dengan alat konvensional dengan Persamaan 6.1
7. Menghitung *error* dari selisih data yang didapat dengan rumus Persamaan 6.2

$$\text{Selisih Data} = \text{Data akuisisi sensor} - \text{Data alat konvensional} \quad (6.1)$$

$$\text{Error Data} = \frac{\text{Selisih Data}}{\text{Data Alat Konvensional}} \times 100\% \quad (6.2)$$

## 3. Hasil Pengujian dan Analisis

Pada proses pengujian mengakuisisi data sensor suhu seperti Gambar 6.6 dilakukan dengan cara mencelupkan sensor suhu dengan termometer suhu konvensional secara bersamaan di air sebanyak 10 kali. Pengujian ini dilakukan seperti pada Gambar 6.3 di bawah ini sesuai dengan prosedur di atas.



**Gambar 6.6 Pengujian Mengakuisisi Data Sensor Suhu**

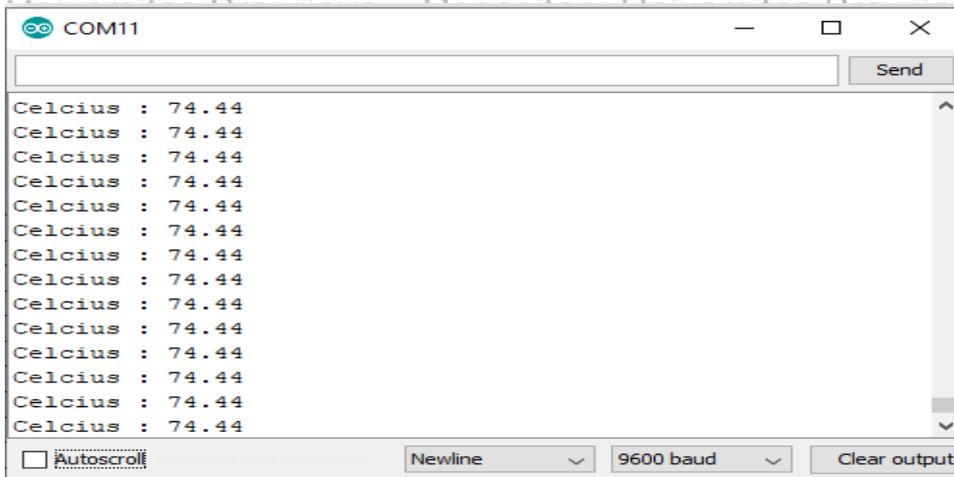
Hasil pengujian sensor suhu yang telah dilakukan dapat diketahui hasil akurasi *error* sensor suhu tersebut sebesar 1.86%. Dengan akurasi *error* 1.86% pada

pengujian tersebut menandakan sensor suhu memiliki akurasi yang sangat baik dan sensor suhu yang digunakan dapat digunakan pada penelitian ini dan selanjutnya. Berikut hasil pengujian sensor suhu seperti pada Tabel 6.2 di bawah ini:

**Tabel 6.2 Tabel Hasil Pengujian Mengakuisisi Data Sensor Suhu**

Pengujian ke	Data Termometer Suhu Konvensional	Data akuisisi sensor Suhu DS18B20	Selisih Data	Error Data
1	9,1 °C	9,75 °C	0,65 °C	7,14%
2	16,3 °C	16,44 °C	0,14 °C	0,85%
3	19,1 °C	19,75 °C	0,65 °C	3,40%
4	21,8 °C	22,62 °C	0,22 °C	1,02%
5	24,9 °C	25,06 °C	0,16 °C	0,64%
6	25,5 °C	25,69 °C	0,19 °C	0,74%
7	32,0 °C	33,06 °C	1,06 °C	3,37%
8	37,1 °C	36,75 °C	0,35 °C	0,94%
9	51,7 °C	51,88 °C	0,18 °C	0,34%
10	74,5 °C	74,44 °C	0,16 °C	0,21%
Rata - rata				1,86%

Hasil dari pengujian sensor pH seperti yang ditampilkan pada Tabel 6.2 diketahui dari *output* sensor suhu pada serial monitor arduino IDE. Setelah melakukan pengujian sensor suhu sebanyak 10 kali terdapat *error* pembacaan akuisisi sensor suhu dikarenakan banyak faktor. Salah satu faktor nya perbedaan dalam kecepatan pembacaan suhu air dari sensor dengan tester suhu air. *Output* sensor suhu tersebut seperti pada Gambar 6.7.



**Gambar 6.7 Hasil Serial Monitor Pengujian Sensor Suhu**



Gambar di atas merupakan salah satu pengujian sensor suhu seperti pada Tabel 6.2 pengujian ke-10 dengan *output* serial monitor Celcius : 74,44 °C. *Output* pada suhu air tester adalah 74,5 °C. Terdapat 0,16% *error* pembacaan sensor suhu yang digunakan.

### 6.2.3 Pengujian Akuisisi Hasil Data Sensor Kedalaman Air

Sensor Kedalaman Air yang digunakan pada penelitian ini sensor JSN-SR04T yang memiliki fungsi kerja dengan menghitung waktu dari *transmitter* memancarkan sinar ultrasonik hingga sinar ultrasonik diterima oleh *receiver*. Dari waktu yang diterima oleh sensor akan dikonversi menjadi jarak halangan yang menjadi pantulan sinar ultrasonik tersebut. Pengujian sensor kedalaman air dengan kode TS-03 dilakukan pada kolam renang yang memiliki kedalaman air yang bervariasi sehingga dapat diketahui akurasi dari sensor yang di uji. Kecepatan rambat sinar ultrasonik pada di dalam air dan di luar air berbeda. Ketika di dalam air kecepatan rambat ultrasonik sebesar 1498 m/s dan untuk di atas air sebesar 343 m/s sehingga kecepatan rambat di dalam air 4,4 kali lebih cepat daripada di atas air. Sensor kedalaman air yang digunakan minimal pembacaan 20 CM di atas air setara 80 CM di dalam air. Sehingga ketika di dalam air minimal pembacaan sensor ini adalah 80 CM.

#### 1. Tujuan Pengujian

Pengujian sensor kedalaman air berguna untuk mengetahui akurasi dari sensor kedalaman air tersebut. Pemilihan kolam renang untuk tempat pengujian dikarenakan kolam renang memiliki kedalaman air yang mempermudah proses pengujian, yaitu dengan diketahui kedalaman kolam renang secara cepat dibandingkan waduk atau bendungan yang perlu ke tengah untuk memastikan kedalaman air dengan meteran.

#### 2. Prosedur Pengujian

Prosedur pengujian sensor kedalaman air seperti pada berikut ini:

1. Melakukan pengukuran kedalaman air pada pinggir kolam renang secara manual dengan meteran agar diketahui nilai kedalaman kolam renang tersebut.
2. Menghubungkan sensor kedalaman air dengan arduino nano melalui pin Digital D7 untuk *ECHO* dan pin Digital D8 untuk *TRIGGER* dan pin *VCC* pada pin 5V *Driver* motor seperti pada Gambar 5.6 Perancangan Skema Sensor Kedalaman JSD-SR04T.
3. Hubungkan arduino nano dengan laptop dan jalankan program yang sudah tersedia dan menyalakan *driver* motor untuk memberikan daya pada sensor.
4. Hasil pengujian dari sensor kedalaman air dapat dilihat pada serial monitor di arduino IDE.
5. Menghitung selisih data sensor dengan alat konvensional dengan persamaan 6.1



$$\text{Selisih Data} = \text{Data akuisisi sensor} - \text{Data alat konvensional} \quad (6.1)$$

6. Menghitung *error* dari selisih data yang didapat dengan rumus persamaan

6.2

$$\text{Error Data} = \frac{\text{Selisih Data}}{\text{Data Alat Konvensional}} \times 100\% \quad (6.2)$$

### 3. Hasil Pengujian dan Analisis

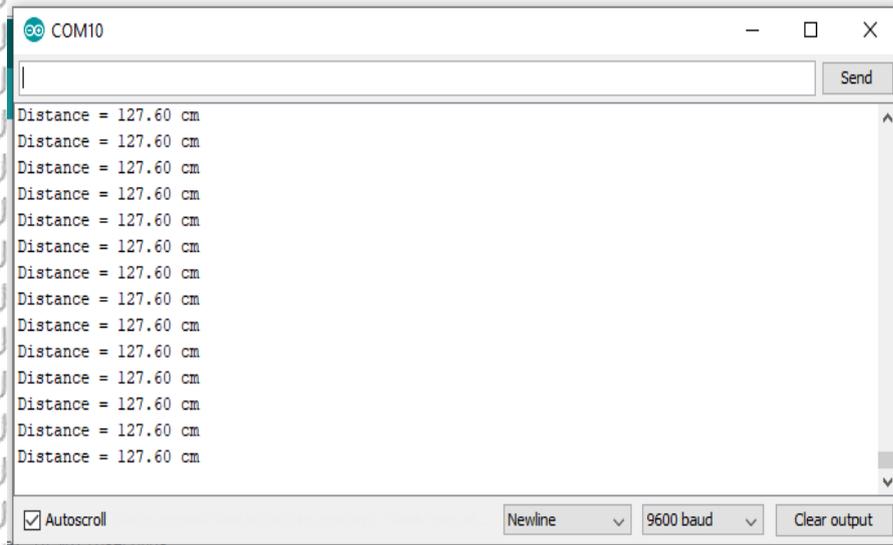
Pengujian sensor kedalaman air dilakukan oleh si peneliti pada kolam renang di Madiun Jawa Timur. Pemilihan lokasi kolam renang dikarenakan pada kolam renang dapat diketahui kedalaman air yang bervariasi dan aman untuk melakukan pengujian. Pemilihan lokasi kolam renang sudah dirasa tepat karena banyak lokasi untuk melakukan pengujian sensor kedalaman seperti pada bendungan air atau waduk tetapi karena faktor *safety* maka dipilihlah kolam renang untuk melakukan pengujian. Kolam renang yang digunakan untuk pengujian memiliki kedalaman sekitar 1 meter hingga 2,5 meter. Dengan kedalaman tersebut maka si peneliti melakukan pengujian di 10 titik pada kolam renang dimulai dari yang terdangkal hingga terdalam.

Pengujian sensor kedalaman air di lokasi kolam renang seperti pada Gambar 6.8 dilakukan di beberapa titik untuk diketahui akurasi dari akuisisi sensor kedalaman air yang digunakan pada penelitian ini. Hasil pengujian didapatkan akurasi *error* dari pembacaan sensor sebesar 2,06 %. Dengan akurasi *error* 2,06% merupakan akurasi *error* yang kecil sehingga sensor kedalaman air yang digunakan memiliki akurasi yang sangat baik.

**Tabel 6.3 Tabel Hasil Pengujian Mengakuisisi Hasil Data Sensor Kedalaman Air**

Pengujian Ke	Kedalaman Air	Hasil Pembacaan Sensor JSN-SR04T	Selisih Data	<i>Error Data</i>
1	125 CM	127,60 CM	2,60 CM	2,08 %
2	135 CM	136,40 CM	1,40 CM	1,03 %
3	150 CM	154 CM	4 CM	2,67 %
4	165 CM	167,20 CM	2,20 CM	1,33 %
5	175 CM	176 CM	1 CM	0,57 %
6	185 CM	189 CM	4 CM	2,16 %
7	205 CM	206,04 CM	1,04 CM	0,51 %
8	210 CM	220 CM	10 CM	4,71 %
9	225 CM	228,80 CM	3,80 CM	1,69 %
10	250 CM	259,60 CM	9,60 CM	3,84 %
Rata-rata				2,06 %

Data hasil proses pengujian seperti pada Tabel 6.4 didapatkan dengan cara mengamati *output* sensor yang ditampilkan pada serial monitor seperti pada Gambar 6.8. Alat pembanding dalam pengujian ini adalah meteran sehingga nilai dari kedalaman tidak terdapat angka dibelakang koma. Untuk sensor kedalaman air terdapat 2 angka di belakang koma karena faktor kecepatan rambat sinar ultrasonik di dalam air lebih cepat 4,4x dari pada di atas air sehingga *output* dari sensor terdapat angka dibelakang koma.



**Gambar 6.8 Hasil Serial Monitor Pengujian Sensor Kedalaman Air**

Gambar di atas merupakan salah satu contoh *output* pengujian ke 1 yang ada pada Tabel 6.3 diatas. Gambar 6.9 menampilkan *output* sensor sebesar 127,60 centimeter. Pada kedalaman air sebenarnya sebesar 125 centimeter sehingga terdapat perbedaan pembacaan sebesar 2,60 centimeter. Setelah pengujian sebanyak 10 titik maka didapatkan hasil perbedaan pembacaan kedalaman air dengan jarak 1 centimeter hingga 9,60 centimeter.

#### **6.2.4 Pengujian Akuisisi Hasil Data Sensor *Turbidity***

Pengujian sensor *turbidity* dilakukan dengan memasukan sensor ke dalam beberapa sampel air yang sudah disiapkan dengan tingkat kekeruhan yang berbeda beda. Pengujian data sensor *turbidity* adalah salah satu pengujian dengan kode TS-04. *Output* sensor *turbidity* yang berupa voltase maka perlu dilakukan konversi nilai voltase menjadi satuan kekeruhan yaitu NTU dengan rumus regresi linier. Rumus regresi linier didapatkan dari data nilai sensor *turbidity* ketika dilakukan kalibrasi sebelum pengujian. Proses kalibrasi dilakukan untuk mendapatkan nilai voltase dari sensor dengan sampel air yang sudah disiapkan sebelumnya.

##### **1. Tujuan Pengujian**

Pengujian sensor *turbidity* memiliki tujuan untuk mengetahui akurasi dari sensor yang digunakan. Sampel air uji yang disiapkan sebelum dilakukan pengujian memiliki tingkat kekeruhan berbeda beda dengan jumlah 5 sampel air. 5 sampel air berisi dari air jernih hingga air keruh dengan tingkat kekeruhan 1000 NTU. Dengan



mengetahui akurasi sensor *turbidity* bisa didapatkan keakuratan tingkat kekeruhan air yang diuji karena kekeruhan air merupakan salah satu elemen penting dalam mengukur kualitas air.

## 2. Prosedur Pengujian

Berikut prosedur pengujian sensor *turbidity* seperti di bawah ini:

1. Menghubungkan sensor *turbidity* dengan arduino nano dengan pin Analog A3, pin *ground* dan pin *VCC* dihubungkan dengan pin 5V *driver* motor seperti pada Gambar 5.7 Perancangan Skema Sensor Kekeruhan *Turbidity* dan setelah itu hubungkan arduino nano dengan laptop.
2. Menjalakan program yang ada pada arduino nano dan nyalakan *driver* motor.
3. Melakukan kalibrasi sensor dengan sampel air kalibrasi dan didapatkan nilai *output* sensor untuk dimasukkan ke dalam rumus regresi linier.
4. Rumus regresi linier yang didapatkan dari proses kalibrasi sensor digunakan untuk proses pengujian selanjutnya.
5. Hasil pengujian dari sensor kedalaman air dapat dilihat pada serial monitor di arduino IDE.
6. Menghitung selisih data sensor dengan alat konvensional dengan persamaan 6.1.

$$\text{Selisih Data} = \text{Data akuisisi sensor} - \text{Data alat konvensional} \quad (6.1)$$

7. Menghitung *error* dari selisih data yang didapat dengan rumus persamaan 6.2.

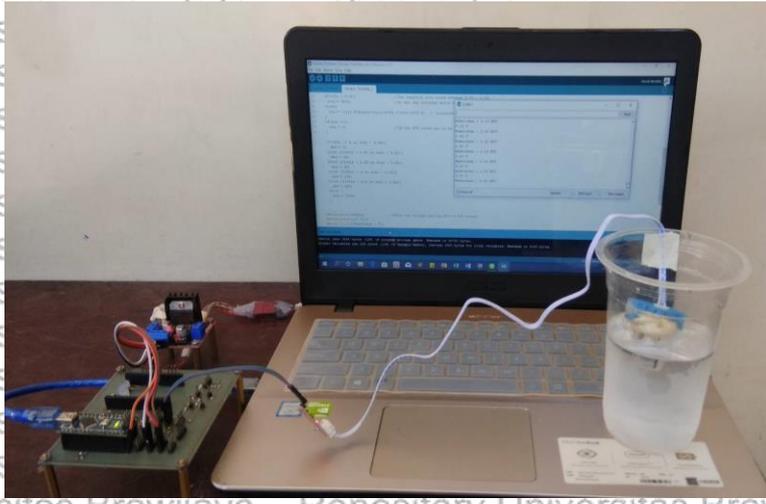
$$\text{Error Data} = \frac{\text{Selisih Data}}{\text{Data Alat Konvensional}} \times 100\% \quad (6.2)$$

## 3. Hasil Pengujian dan Analisis

Pengujian sensor kekeruhan air dilakukan untuk mengetahui sensor yang digunakan dapat melakukan akuisisi data sesuai dengan air yang di uji. Pada pengujian ini si peneliti melakukan pengamatan *output* dari sensor berupa voltase apakah terjadi perubahan atau tidak ketika di uji dengan air dengan tingkat kekeruhan yang berbeda beda. Gambar 6.9 merupakan contoh pengujian sensor kekeruhan air dengan beberapa sampel air keruh yang berbeda beda.

Pengujian sensor kekeruhan air dilakukan sebanyak 5 kali sesuai dengan prosedur. Hasil pengujian yang telah dilakukan seperti pada Tabel 6.4 di bawah ini. Dengan melakukan pengujian tersebut si peneliti dapat mengetahui akurasi dari sensor kekeruhan yang digunakan dan dapat menjawab rumusan masalah pada bab

1. Hasil pengujian juga bisa menjadi referensi untuk penelitian selanjutnya atau untuk penelitian lain dengan berhubungan sensor kekeruhan yang digunakan pada penelitian ini.

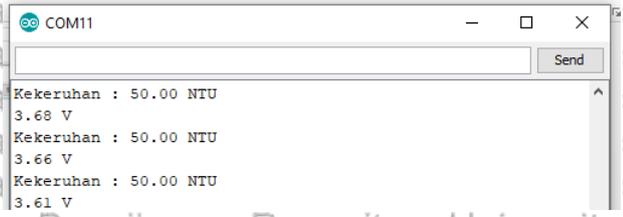


Gambar 6.9 Hasil Pengujian Mengakuisisi Hasil Data Sensor *Turbidity*

Tabel 6.4 Tabel Hasil Pengujian Mengakuisisi Data Sensor Kekeruhan Air

Pengujian Ke-	Kondisi Air	Nilai Sensor
1	Air Jernih	4,86 V
2	Ditambahkan 1 Sendok Tanah	4,13 V
3	Ditambahkan 2 Sendok Tanah	3,61 V
4	Ditambahkan 3 Sendok Tanah	2,93 V
5	Ditambahkan 4 Sendok Tanah	1,80 V

Hasil pengujian yang sesuai prosedur seperti Tabel 6.4 di atas diketahui nilai sensor berupa voltase dan *Output*. Pada kolom *Output* terdapat nilai sensor yang menandakan bahwa sensor yang digunakan ketika dilakukan pengujian dengan air keruh yang berbeda maka akan menghasilkan *output* yang berbeda pula dan linier. Semakin keruh air yang dideteksi oleh sensor maka sensor akan menghasilkan *output* voltase yang semakin kecil pula dan *output* nya linier. Hal ini terjadi karena semakin keruh air maka akan membuat sensor pada bagian *receiver* semakin kecil menerima sinar dari *transmitter*. *Output* berupa voltase dari sensor kekeruhan akan dimasukkan ke rumus regresi linier untuk menentukan *output* NTU. Gambar 6.11 merupakan contoh hasil *output* pengujian ke 3 pada Tabel 6.5 di atas dengan voltase 3,61. Untuk hasil konversi ke rumus regresi linier adalah senilai 50 NTU.



Gambar 6.10 Hasil Serial Monitor Pengujian Sensor Kekeruhan Air



### 6.2.5 Pengujian Akuisisi Hasil Data Sensor TDS

Kode TS-05 adalah pengujian akuisisi hasil data sensor TDS dilakukan sebanyak 10 kali agar diketahui nilai data hasil akuisisi sensor TDS yang menjadi salah satu data dalam klasifikasi metode k-NN. Sampel air yang digunakan dalam pengujian sensor TDS ini memiliki nilai PPM yang berbeda-beda mulai dari nilai terendah hingga tertinggi. Akurasi dari data hasil akurasi sensor TDS yang digunakan dapat diketahui dengan membandingkan data tersebut dengan TDS tester konvensional dan dengan membandingkan nilai tersebut dapat diketahui selisih *error* nya.

#### 1. Tujuan Pengujian

Pengujian ini dilakukan dengan tujuan agar dapat diketahui akurasi dari sensor TDS yang digunakan. Dengan diketahui akurasi dari sensor tersebut bisa dijadikan patokan bahwa sensor yang digunakan akurat atau tidak. Proses pengujian dilakukan dengan cara menaruh sensor TDS dan TDS tester konvensional secara bersamaan pada sampel air yang sudah disiapkan. Sampel air yang digunakan berjumlah 10 dengan nilai PPM yang berbeda-beda sehingga diketahui nilai akurasi dari sensor ketika dilakukan pengujian.

#### 2. Prosedur Pengujian

Prosedur pengujian sensor TDS seperti pada berikut ini:

1. Sensor TDS dilakukan pengujian sebanyak sepuluh kali dengan membandingkan TDS tester dengan sampel air yang memiliki nilai PPM berbeda-beda.
2. Menghubungkan laptop dengan *microcontroller* arduino nano dan hubungkan sensor TDS dengan arduino nano melalui pin Analog A2, pin *ground* dan pin *VCC* dihubungkan dengan *driver* motor 5V sesuai dengan Gambar 5.8 Perancangan Skema Sensor TDS.
3. Jalankan program pada arduino nano dan nyalakan *driver* motor untuk memberikan daya sensor TDS.
4. Hasil pengujian dari sensor TDS dapat dilihat pada serial monitor di arduino IDE.
5. Menghitung selisih data sensor dengan alat konvensional dengan Persamaan 6.1.

$$\text{Selisih Data} = \text{Data akuisisi sensor} - \text{Data alat konvensional} \quad (6.1)$$

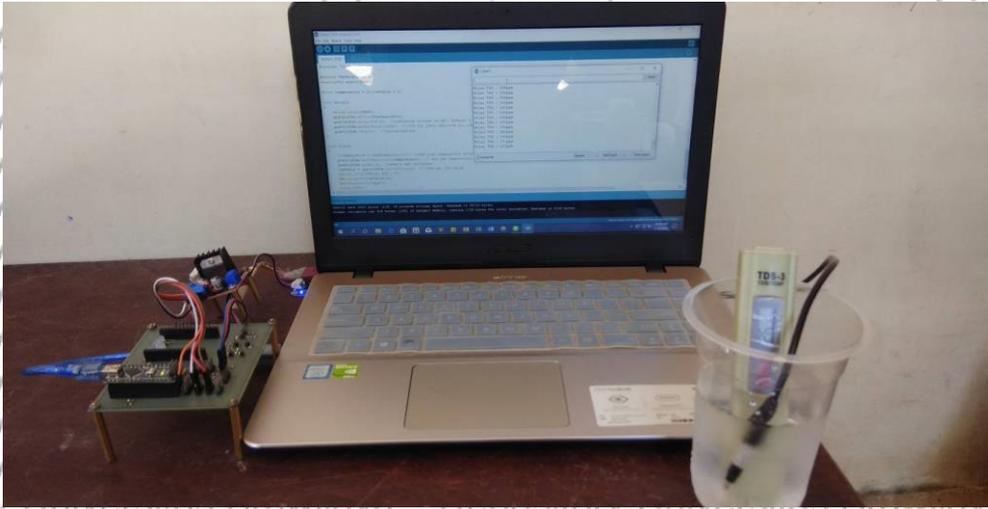
6. Menghitung *error* dari selisih data yang didapat dengan rumus Persamaan 6.2.

$$\text{Error Data} = \frac{\text{Selisih Data}}{\text{Data Alat Konvensional}} \times 100\% \quad (6.2)$$



### 3. Hasil Pengujian dan Analisis

Sensor TDS yang dilakukan proses pengujian seperti Gambar 6.12 dengan cara memasukan sensor tersebut dengan TDS tester secara bersamaan pada sampel air yang sudah disiapkan yang berjumlah 10 sampel air. Pengujian ini digunakan untuk mengetahui akurasi dari sensor TDS yang digunakan pada penelitian ini. Berikut tabel di bawah ini hasil pengujian sensor TDS seperti Tabel 6.5:



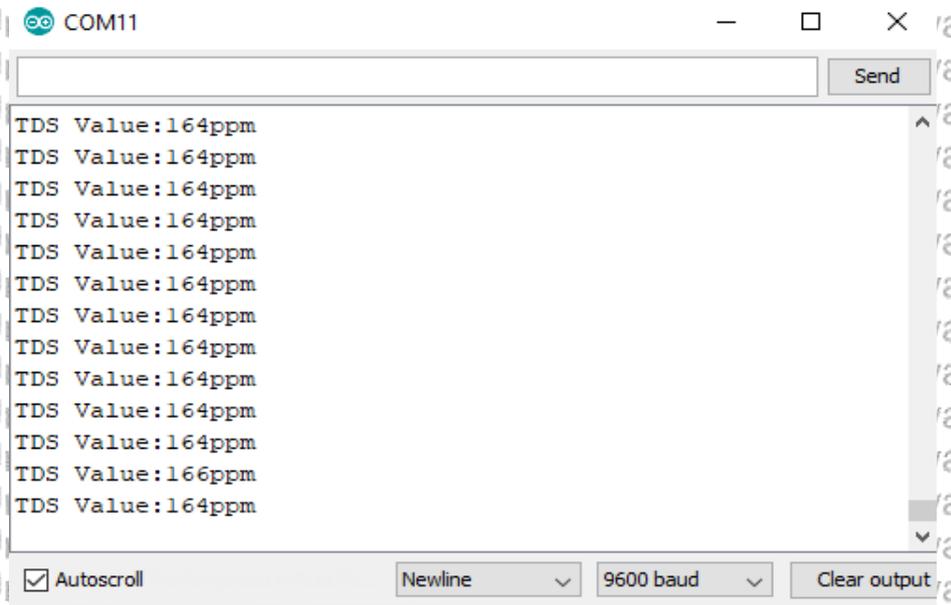
**Gambar 6.11 Pengujian Akuisisi Hasil Data Sensor TDS**

**Tabel 6.5 Tabel Hasil Pengujian Mengakuisisi Hasil Data Sensor TDS**

Pengujian ke-	TDS Tester Konvensional	Sensor TDS DF-ROBOT	Selisih Data	Error Data
1	17 PPM	20 PPM	3 PPM	17,64%
2	155 PPM	164 PPM	9 PPM	6,33%
3	174 PPM	181 PPM	7 PPM	4,02%
4	190 PPM	211 PPM	11 PPM	5,78%
5	246 PPM	232 PPM	14 PPM	5,62%
6	278 PPM	290 PPM	12 PPM	4,31%
7	367 PPM	382 PPM	15 PPM	4,10%
8	460 PPM	483 PPM	22 PPM	4,78%
9	531 PPM	548 PPM	17 PPM	3,20%
10	1030 PPM	1000 PPM	30 PPM	2,91%
Rata-rata				5,86%

Hasil data akuisisi sensor TDS dalam proses pengujian seperti pada Tabel 6.5 merupakan data yang ditampilkan pada serial monitor arduino IDE. Akurasi error dari data hasil akuisisi sensor TDS sebesar 5,86%. Perbedaan pada sensor TDS yang

digunakan karena pada label produk terdapat akurasi perbedaan  $\pm 10\%$  sehingga membuat pembacaan sensor TDS terdapat perbedaan. Akurasi sensor TDS yang sudah diketahui bisa menjadikan sensor TDS yang digunakan memiliki akurasi yang baik sehingga sensor TDS ini dapat digunakan dalam penelitian selanjutnya.



**Gambar 6.12 Hasil Serial Monitor Pengujian Sensor TDS**

Pengujian sensor TDS seperti pada Gambar 6.13 merupakan pengujian ke-2 pada Tabel 6.5 yang melakukan akuisisi data sensor pada air dengan jumlah zat terlarut sebanyak 164 PPM. Nilai dari TDS tester pada pengujian air yang sama menunjukkan nilai 155 PPM dan terjadi perbedaan yang cukup besar dengan error 6.33%. Hal tersebut terjadi dengan banyak faktor kemungkinan seperti perbedaan pembacaan suhu pada sensor TDS dan TDS tester. Pada sensor TDS yang digunakan menggunakan variabel suhu air sebagai salah satu menentukan jumlah zat yang terlarut.

### **6.3 Pengujian Sistem dengan Metode k-NN**

Nilai K pada klasifikasi k-NN merupakan salah satu element yang paling penting dalam menentukan akurasi dari klasifikasi k-NN. Metode k-NN menggunakan nilai K dalam proses mengambil sejumlah K data yang paling dekat dari proses *euclidean*, sehingga nilai K sangat berpengaruh besar sekali dalam klasifikasi k-NN. Nilai K biasanya bernilai ganjil dan nilai K yang diuji bernilai 5,7,9,11. Proses pengujian dengan kode TS-06 dilakukan dengan cara menjadikan nilai K 5,7,9,11 secara bergantian dalam proses klasifikasi k-NN.

#### **6.3.1 Tujuan Pengujian**

Pengujian akurasi dari pemilihan nilai K memiliki tujuan agar diketahui nilai K yang paling optimal digunakan dalam klasifikasi k-NN. Selain mengetahui nilai K yang paling optimal, proses pengujian pemilihan nilai K bisa mengetahui hasil klasifikasi dari metode k-NN dalam penelitian ini.



### 6.3.2 Prosedur Pengujian

1. Menyambungkan node MCU dengan laptop dan menyalakan *driver* motor untuk memberikan daya untuk kelima sensor yang digunakan.
2. Menjalankan program klasifikasi k-NN yang ada pada node MCU dan program akuisisi data pada arduino nano. Proses klasifikasi dijalankan pada node MCU dan arduino nano digunakan untuk mengakuisisi data sensor yang digunakan.
3. Melakukan pemilihan nilai K yang bernilai 5, 7, 9, 11 secara bergantian dalam proses klasifikasi k-NN.
4. Melihat hasil klasifikasi k-NN dengan nilai K 5, 7, 9, 11 pada serial monitor arduino IDE.
5. Melakukan pemilihan nilai K yang paling optimal dengan melihat dari akurasi k-NN di tiap tiap nilai K yang diuji dengan perhitungan melalui Persamaan 6.3.

$$Akurasi = \frac{Total\ semua\ data - Total\ data\ yang\ tidak\ sesuai}{Total\ semua\ data} \times 100 \quad (6.3)$$

### 6.3.3 Hasil Pengujian

Setelah melakukan pengujian sesuai dengan prosedur di atas akan didapatkan akurasi dari klasifikasi k-NN. Dengan variasi nilai K pada pengujian ini diharapkan peneliti dapat menentukan pemilihan nilai K yang paling optimal. Hasil pengujian akurasi dari pemilihan nilai K seperti pada Tabel 6.6.

**Tabel 6.6 Tabel Hasil Pengujian Akurasi dari Pemilihan Nilai K**

No	pH	Suhu	Kekeruhan	Kedalaman	TDS	Kelas	K = 5	K = 7	K = 9	K = 11
1	8,45	25,06 °C	0,1 NTU	250,8 CM	91 PPM	Bagus	Ba gu s	Ba gu s	Ba gu s	Ba gu s
2	7,9	24,3 °C	1 NTU	211,2 CM	220 PPM	Bagus	Ba gu s	Ba gu s	Ba gu s	Ba gu s
3	7,1	25,2 °C	0,1 NTU	220,8 CM	180 PPM	Bagus	Ba gu s	Ba gu s	Ba gu s	Ba gu s
4	8,15	26 °C	0,1 NTU	272,8 CM	150 PPM	Bagus	Ba gu s	Ba gu s	Ba gu s	Ba gu s
5	7,5	24,6 °C	0,1 NTU	250,8 CM	245 PPM	Bagus	Ba gu s	Ba gu s	Ba gu s	Ba gu s
6	7,63	25,7 °C	0,1 NTU	240,9 CM	139 PPM	Bagus	Ba gu s	Ba gu s	Ba gu s	Ba gu s
7	7,82	24,95 °C	0,1 NTU	294,8 CM	100 PPM	Bagus	Ba gu s	Ba gu s	Ba gu s	Ba gu s



8	7,16	24,81 °C	10 NTU	294,8 CM	101 PPM	Bagus	Ba gus	Ba gus	Ba gus	Ba gus
9	8,25	23 °C	10 NTU	272,8 CM	198 PPM	Bagus	Ba gus	Ba gus	Ba gus	Ba gus
10	7,18	23,43 °C	0,1 NTU	211,2 CM	99 PPM	Bagus	Ba gus	Ba gus	Ba gus	Ba gus
11	8,41	25,88 °C	50 NTU	272,8 CM	300 PPM	Sedang	Se da ng	Se da ng	Se da ng	Se da ng
12	8,21	24,9 °C	100 NTU	294,8 CM	315 PPM	Sedang	Se da ng	Se da ng	Se da ng	Se da ng
13	8,8	26,19 °C	50 NTU	211,2 CM	340 PPM	Sedang	Se da ng	Se da ng	Se da ng	Se da ng
14	8,69	25,13 °C	100 NTU	220 CM	354 PPM	Sedang	Se da ng	Se da ng	Se da ng	Se da ng
15	8,14	25,43 °C	100 NTU	272,8 CM	311 PPM	Sedang	Se da ng	Se da ng	Se da ng	Se da ng
16	8,58	26,21 °C	50 NTU	240,9 CM	283 PPM	Sedang	Se da ng	Se da ng	Se da ng	Se da ng
17	7,91	24,5 °C	50 NTU	211,2 CM	256 PPM	Sedang	Se da ng	Se da ng	Se da ng	Se da ng
18	7,84	24,13 °C	50 NTU	240,9 CM	241 PPM	Sedang	Ba gus	Ba gus	Ba gus	Ba gus
19	8,96	26,31 °C	100 NTU	294,8 CM	329 PPM	Sedang	Se da ng	Se da ng	Se da ng	Se da ng
20	8,58	25,89 °C	100 NTU	294,8 CM	350 PPM	Sedang	Se da ng	Se da ng	Se da ng	Se da ng
21	5,99	26,4 °C	200 NTU	272,8 CM	430 PPM	Jelek	Jel ek	Jel ek	Jel ek	Se da ng



22	8,9	28,1 °C	200 NTU	240,8 CM	398 PPM	Jelek	Se da ng	Se da ng	Se da ng	Se da ng
23	6,2	27,5 °C	100 NTU	176 CM	453 PPM	Jelek	Jel ek	Jel ek	Jel ek	Jel ek
24	6,5	28,9 °C	100 NTU	211,2 CM	517 PPM	Jelek	Jel ek	Jel ek	Jel ek	Jel ek
25	7	28,36 °C	200 NTU	211,2 CM	381 PPM	Jelek	Jel ek	Jel ek	Jel ek	Se da ng
Akurasi							92 %	92 %	92 %	86 %

Tabel 6.6 merupakan hasil pengujian klasifikasi k-NN dengan 4 macam nilai K. 4 macam nilai k menghasilkan hasil klasifikasi yang berbeda beda. Untuk nilai K = 5, 7, 9 terdapat 2 data yang tidak sesuai dengan hasil kelas yang sebenarnya. Untuk nilai K = 11 terdapat 4 data yang tidak sesuai dengan hasil kelas yang sebenarnya. Dari hasil tersebut maka nilai K=5 digunakan untuk klasifikasi k-NN pada penelitian ini karena dengan nilai K=5 menghasilkan akurasi yang paling baik dari nilai K yang lainnya.

Perhitungan akurasi dari data hasil klasifikasi menggunakan Persamaan 6.3 sebagai berikut:

$$Akurasi (K3) = \frac{Total\ semua\ data - Total\ data\ yang\ tidak\ sesuai}{Total\ semua\ data} \times 100$$

$$Akurasi = \frac{25 - 2}{25} \times 100$$

$$Akurasi = \frac{23}{25} \times 100$$

$$Akurasi = 92\%$$

Contoh pengujian proses klasifikasi k-NN ini menggunakan data sensor yang terdiri dari pH : 7,91, Suhu : 24,5 C, Kekeruhan : 50 NTU, Kedalaman : 211,2 CM, TDS : 256 PPM. Nilai tersebut berasal dari sampel data uji ke 17 pada Tabel 6.6 yang akan dihitung secara manual dengan proses perhitungan manual klasifikasi k-NN seperti di bawah ini.

Contoh proses perhitungan data uji dengan data latih ke-1 dengan rumus *euclidean* seperti persamaan 2.1 di bawah ini. Pada proses klasifikasi k-NN di penelitian ini menggunakan normalisasi min-max karena nilai dari tiap parameter yang digunakan bervariasi dan dengan *range* yang berbeda beda.

$$D = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2 + (x_4 - y_4)^2 + (x_5 - y_5)^2}$$



$$D = \sqrt{(7,91 - 6,11)^2 + (24,5 - 25,13)^2 + (50 - 100)^2 + (211,2 - 211,2)^2 + (256 - 410)^2}$$

$$D(\text{Normalisasi}) = \sqrt{(0,565 - 0,436)^2 + (0,288 - 0,296)^2 + (0,05 - 0,1)^2 + (0,469 - 0,469)^2 + (0,256 - 0,410)^2}$$

$$D(\text{normalisasi}) = \sqrt{(0,129)^2 + (-0,008)^2 + (-0,05)^2 + (0)^2 + (-0,154)^2}$$

$$D(\text{normalisasi}) = \sqrt{0,016641 + 0,00064 + 0,00250 + 0 + 0,02371}$$

$$D(\text{normalisasi}) = 0,020854$$

Maka hasil dari proses perhitungan data uji dengan data latih ke-1 adalah 0,20854 yang sesuai dengan nilai pada Tabel 6.7 pada kolom Data Ke-1 di bawah ini.

Hasil proses perhitungan dengan rumus *euclidean* pada klasifikasi k-NN dapat dilihat seperti pada Tabel 6.7.

**Tabel 6.7** Tabel Hasil Proses Perhitungan *Euclidean*

Hasil Proses Perhitungan <i>Euclidean</i>		
Data Ke	Hasil Nilai	Kelas Kualitas Air
1	0,208	Jelek
2	0,361	Jelek
3	0,333	Jelek
4	0,278	Jelek
5	0,447	Jelek
6	0,390	Jelek
7	0,359	Jelek
8	0,415	Jelek
9	0,302	Jelek
10	0,648	Jelek
11	0,193	Sedang
12	0,222	Sedang
13	0,194	Sedang
14	0,150	Sedang
15	0,180	Sedang
16	0,090	Sedang
17	0,069	Sedang



18	0,297	Sedang
19	0,209	Sedang
20	0,181	Sedang
21	0,203	Sedang
22	0,056	Sedang
23	0,110	Sedang
24	0,099	Sedang
25	0,145	Sedang
26	0,104	Sedang
27	0,228	Sedang
28	0,187	Sedang
29	0,076	Sedang
30	0,181	Sedang
31	0,171	Bagus
32	0,163	Bagus
33	0,113	Bagus
34	0,240	Bagus
35	0,117	Bagus
36	0,232	Bagus
37	0,171	Bagus
38	0,087	Bagus
39	0,201	Bagus
40	0,217	Bagus
41	0,219	Bagus
42	0,281	Bagus
43	0,269	Bagus
44	0,228	Bagus
45	0,222	Bagus
46	0,145	Bagus
47	0,058	Bagus
48	0,121	Bagus



49	0,153	Bagus
50	0,120	Bagus

Setelah didapatkan hasil perhitungan data uji dengan data latih menggunakan rumus *euclidean*, langkah selanjutnya melakukan sorting *ascending* (dimulai terkecil) hingga didapatkan data seperti pada Tabel 6.8 di bawah ini.

**Tabel 6.8 Tabel Hasil Data Setelah *Sorting***

Hasil Data Setelah <i>Sorting</i>			
Urutan	Data Ke	Hasil Nilai	kualitas
1	22	0,056	Sedang
2	47	0,058	Bagus
3	17	0,069	Sedang
4	29	0,076	Sedang
5	38	0,087	Bagus
6	16	0,090	Sedang
7	24	0,099	Sedang
8	26	0,104	Sedang
9	23	0,110	Sedang
10	33	0,113	Bagus
11	35	0,117	Bagus
12	50	0,120	Bagus
13	48	0,121	Bagus
14	25	0,145	Sedang
15	46	0,145	Bagus
16	14	0,150	Sedang
17	49	0,153	Bagus
18	32	0,163	Bagus
19	31	0,171	Bagus
20	37	0,171	Bagus
21	15	0,180	Sedang
22	30	0,180	Sedang
23	20	0,181	Sedang



24	28	0,187	Sedang
25	11	0,193	Sedang
26	13	0,194	Sedang
27	39	0,201	Bagus
28	21	0,203	Sedang
29	1	0,207	Jelek
30	19	0,209	Sedang
31	40	0,217	Bagus
32	41	0,219	Bagus
33	45	0,222	Bagus
34	12	0,222	Sedang
35	27	0,228	Sedang
36	44	0,228	Bagus
37	36	0,232	Bagus
38	34	0,240	Bagus
39	43	0,269	Bagus
40	4	0,278	Jelek
41	42	0,281	Bagus
42	18	0,297	Sedang
43	9	0,302	Jelek
44	3	0,333	Jelek
45	7	0,359	Jelek
46	2	0,361	Jelek
47	6	0,390	Jelek
48	8	0,415	Jelek
49	5	0,447	Jelek
50	10	0,648	Jelek

Langkah selanjutnya melakukan pengambilan data sejumlah  $K = 5,7,9,11$  dan dilakukan proses *weighted voting*. *Weighted voting* merupakan metode *voting* dalam klasifikasi k-NN dengan cara menghitung bobot dari tiap-tiap hasil klasifikasi dan dijumlahkan dengan sesama kualitas data tersebut. Nilai yang lebih besar hasil dari metode *weighted voting* akan diambil menjadi hasil klasifikasi.

Tabel 6.9 Tabel Hasil *Sorting* dengan Nilai K

Urutan	Data Ke -	Hasil Nilai	Kualitas
1	22	0,056	Sedang
2	47	0,058	Bagus
3	17	0,069	Sedang
4	29	0,076	Sedang
5	38	0,087	Bagus
6	16	0,090	Sedang
7	24	0,099	Sedang
8	26	0,104	Sedang
9	23	0,110	Sedang
10	33	0,113	Bagus
11	35	0,117	Bagus

Rumus metode *weighted voting* seperti persamaan .

$$votes(Kualitas) = \frac{1}{d(data,kualitas)^2} \approx Hasil Klasifikasi$$

Karena dalam K data hasil proses *euclidean* yang paling terdekat dari data uji hanya terdapat klasifikasi sedang dan bagus maka klasifikasi tersebut yang dilakukan metode *weighted voting*.

Tabel 6.10 Hasil Perhitungan *Weighted Voting*

	K = 5	K = 7	K = 9	K = 11
Vote Sedang	708,1510	928,5358	1102,6351	1102,6351
Vote Bagus	430,5216	430,5216	430,5216	460,1254

Berikut contoh perhitungan *weighted voting* untuk K = 5 :

$$votes(Sedang(K = 5)) = \frac{1}{d(0,056)^2} + \frac{1}{d(0,069)^2} + \frac{1}{d(0,076)^2} \approx 708,1510$$

$$votes(Bagus(K = 5)) = \frac{1}{d(0,058)^2} + \frac{1}{d(0,087)^2} \approx 430,5216$$

Dari hasil yang didapatkan nilai *vote* sedang lebih besar dari pada *vote* bagus sehingga hasil klasifikasi dari k-NN dengan data pH : 7,91, Suhu : 24,5 °C, Kekeruhan : 50 NTU, Kedalaman : 211,2 CM, TDS : 256 PPM adalah air yang memiliki kualitas SEDANG dengan K = 5.



## 6.4 Pengujian Waktu Komputasi Sistem dan Penggunaan *Memory*

Pengujian waktu komputasi sistem digunakan untuk mengetahui sistem yang dibuat memerlukan waktu seberapa lama untuk melakukan klasifikasi kualitas air dengan metode k-NN. Bentuk pengujian waktu komputasi berupa waktu komputasi pada node MCU ketika melakukan akuisisi data kelima sensor ditambah dengan waktu komputasi dari node MCU dimulai dari menerima data pada arduino nano lalu diproses klasifikasi k-NN hingga diakhiri proses mengirim data menuju firebase. Pada pengujian ini menggunakan koneksi WiFi dari *handphone* dengan jarak *range* dengan alat sekitar 1 meter. Selain itu pada pengujian ini mendapatkan informasi penggunaan *memory* pada arduino nano dan node MCU. Sistem yang dibuat pada memerlukan *memory* seberapa besar untuk arduino nano dan node MCU. Dengan mengetahui penggunaan *memory* maka hasil pengujian ini dapat menjadi referensi penelitian lain.

### 6.4.1 Tujuan

Tujuan dari pengujian ini diharapkan si peneliti bisa mengetahui waktu rata-rata dari sistem yang dibuat dan penggunaan *memory* pada *microcontroller*. Proses pengujian dilakukan sebanyak 10 kali untuk mengukur waktu rata-rata yang diperlukan sistem untuk melakukan sebuah proses klasifikasi hingga selesai dan mengukur kapasitas penggunaan *memory* dari *microcontroller* yang digunakan. Waktu komputasi yang diketahui bisa digunakan sebagai parameter waktu keseluruhan sistem yang dibuat hingga mengirim data menuju firebase.

### 6.4.2 Prosedur

1. Menyalakan *driver* motor untuk memberikan daya kepada *microcontroller* arduino nano, node MCU dan kelima sensor yang digunakan.
2. Menghubungkan *microcontroller* dengan laptop dan menjalankan waktu yang sudah disiapkan.
3. Mencelupkan alat yang dibuat ke wadah air atau bak air.
4. Melihat waktu komputasi dari sistem melalui serial monitor arduino IDE.
5. Mencatat waktu komputasi dari setiap pengujian sistem.
6. Menghitung rata-rata waktu komputasi menggunakan persamaan 6.4

$$\text{Rata - rata} = \frac{\text{Total waktu pengujian}}{\text{Jumlah data pengujian}} \quad (6.4)$$

7. Menghitung penggunaan *memory* pada arduino nano dan node MCU yang dapat dilihat pada aplikasi arduino IDE.

### 6.4.3 Hasil Pengujian dan Analisis

Pengujian waktu komputasi sistem pada penelitian ini adalah gabungan dari waktu komputasi arduino nano dan waktu komputasi node MCU. Hal tersebut karena pada penelitian ini menggunakan 2 *microcontroller* yaitu arduino nano untuk melakukan akuisisi data 5 sensor dan node MCU untuk melakukan proses klasifikasi k-NN dan

mengirim data menuju firebase serta meminta data dari firebase untuk pergerakan alat di atas air. Berikut hasil pengujian waktu komputasi sistem seperti pada Tabel 6.11.

**Tabel 6.11 Tabel Hasil Pengujian Waktu Komputasi Sistem**

Pengujian Ke	Waktu Komputasi Arduino Nano	Waktu Komputasi Node MCU	Waktu Komputasi Keseluruhan (ms)
1	1026 ms	4392 ms	5418 ms
2	1028 ms	2583 ms	3611 ms
3	1025 ms	4442 ms	5467 ms
4	1025 ms	2572 ms	3600 ms
5	1028 ms	2659 ms	3687 ms
6	1025 ms	2621 ms	3643 ms
7	935 ms	2627 ms	3562 ms
8	1026 ms	3572 ms	4598 ms
9	1028 ms	3369 ms	4297 ms
10	1026 ms	2460 ms	3486 ms
	Rata - rata		4135 ms

Dari hasil pengujian seperti pada Tabel 6.11 terdapat waktu rata-rata total yang diperlukan untuk sistem melakukan 1 kali proses hingga selesai adalah 4135 ms. Waktu tersebut didapatkan dari kedua *microcontroller* di mana arduino nano memiliki rata-rata waktu komputasi sebesar 1017 ms dan untuk waktu komputasi Node MCU sebesar 3491 ms. Untuk waktu komputasi node MCU memiliki perbedaan yang cukup banyak antar pengujian dikarenakan node MCU melakukan proses mengirim data menuju firebase dan meminta data dari firebase melalui koneksi WiFi. Lamanya waktu komputasi untuk node MCU karena pada koneksi WiFi dari data *handphone* banyak sekali faktor seperti terdapat aplikasi yang berjalan di belakang layar *handphone* sehingga memerlukan koneksi internet yang lebih. Hal tersebut memungkinkan variasinya waktu komputasi dari node MCU dalam melakukan sebuah proses mengirim data hingga selesai.



**Gambar 6.13 Penggunaan Memory pada Arduino Nano**



```
Done compiling.
```

```
Sketch uses 368864 bytes (35%) of program storage space. Maximum is 1044464 bytes.
Global variables use 30716 bytes (37%) of dynamic memory, leaving 51204 bytes for local variables. Maximum is 81920 bytes.
```

**Gambar 6.14 Penggunaan Memory pada Node MCU**

Memory yang dibutuhkan untuk program agar bisa berjalan pada arduino nano seperti Gambar 6.14 sebesar 13870 bytes dari 30720 bytes. Alokasi memory tersebut adalah besar alokasi memory yang diperlukan untuk menyimpan code yang akan dijalankan. Selain itu program pada arduino nano memerlukan alokasi sebesar 463 bytes dari 2048 bytes untuk global variabel. Global variabel adalah alokasi memory yang digunakan untuk variabel yang di inisialisasi di luar fungsi. Untuk program pada node MCU seperti Gambar 6.15 memerlukan alokasi memory sebesar 368864 bytes dari 1044464 bytes. Alokasi memory tersebut biasanya disimpan pada Flash Memory. Untuk alokasi global variabel pada node MCU memerlukan sebesar 30716 bytes dari 81920 bytes. Global variabel pada biasanya disimpan pada SRAM (Static Random Access Memory) microcontroller yang digunakan.

Pada pengujian ini dapat dilihat contoh pengujian ke 4 pada Tabel 6.11 tersebut untuk arduino nano memiliki waktu komputasi 1025 ms seperti pada Gambar 6.16 dan untuk node MCU memiliki waktu komputasi 2572 ms seperti pada Gambar 6.17. Waktu komputasi untuk arduino nano dihitung dari melakukan akuisisi data 5 sensor dan proses mengirim data menuju node MCU. Untuk node MCU waktu komputasinya dihitung dimulai dari menerima data dari arduino nano lalu melakukan proses klasifikasi k-NN. Setelah itu node mengirim data 5 sensor dan hasil klasifikasi serta meminta data dari firebase untuk menggerakkan motor DC pada alat.

```
{"ph":6.17,"suhu":28.53,"kedalaman":180.80,"kekeruhan":200.00,"TDS":482.00,"counter":38}
WAKTU : 1.0250
```

**Gambar 6.15 Pengujian Waktu Komputasi Arduino Nano**

```
Nilai Indeks
Nilai JELEK : 198.9980164
Nilai SEDANG : 30.5502071
Nilai BAGUS : 0.0000000
-----
HASIL : JELEK
pH : 6.17
Suhu : 28.53
Kedalaman : 180.80
Kekeruhan : 200.00
TDS : 482.00
Gerak Kapal :
WAKTU : 2.5720
```

**Gambar 6.16 Pengujian Waktu Komputasi Node MCU**



## 6.5 Pengujian *Reliabilitas* Proses Pengiriman Data ke *Firestore Database*

Pengujian *reliabilitas* proses pengiriman data ke *Firestore Database* merupakan sebuah pengujian dengan kode TS-07 dan TS-08 yang mencocokkan data hasil akuisisi sensor dan data klasifikasi dari *microcontroller* yang dikirim menuju *Firestore* lalu ditampilkan di aplikasi android. Proses pengujian dilakukan sebanyak 10 kali dengan data akuisisi sensor dan data klasifikasi yang berbeda beda. Pengujian ini dilakukan dengan melihat data melalui serial monitor arduino ide untuk data akuisisi sensor dan data klasifikasi k-NN dan melihat *Firestore* melalui web dan melihat data di aplikasi arduino.

### 6.5.1 Tujuan

Pengujian *reliabilitas* proses pengiriman data ke *Firestore Database* ini memiliki tujuan untuk mengetahui tingkat kehandalan pada saat proses mengirim data yang dilakukan oleh *microcontroller* menuju *Firestore* dan proses aplikasi *smartphone* menampilkan data tersebut. Dengan pengujian ini bisa diketahui ketika proses pengiriman data yang dilakukan oleh node MCU apakah terdapat data yang hilang atau rusak atau tidak. Dengan melakukan pengujian ini diharapkan peneliti bisa mengetahui dan menjawab rumusan masalah tentang tingkat *reliabilitas* pada proses pengiriman dari sistem yang dibuat.

### 6.5.2 Prosedur

1. Menyalakan *driver* motor untuk memberikan daya arduino nano dan kelima sensor.
2. Menyambungkan node MCU dengan laptop dan menjalankan program yang sudah tersedia.
3. Pastikan arduino nano dan node MCU sudah saling terhubung menggunakan UART (RX TX) untuk proses mengirim data dari arduino nano menuju node MCU.
4. Melihat data akuisisi sensor dari arduino nano menuju node MCU dan data hasil klasifikasi di node MCU melalui serial monitor arduino ide.
5. Membuka halaman web *Firestore* dan membuka aplikasi android.
6. Melakukan pengamatan data yang berada di serial monitor, *Firestore* dan aplikasi android apakah terdapat perbedaan atau tidak.

### 6.5.3 Hasil Pengujian dan Analisis

Pengujian ini dilakukan dengan melihat data melalui serial monitor arduino ide untuk data akuisisi sensor dan data klasifikasi k-NN serta melihat *Firestore* melalui web dan melihat data di aplikasi arduino. Pengujian tersebut dilakukan sebanyak 10 kali dan setelah melakukan pengujian maka didapatkan hasil pengujian seperti pada Tabel 6.12 di bawah ini.

**Tabel 6.12 Tabel Hasil Pengujian Proses Pengiriman Data Firebase Database**

No	Data pH	Data Suhu	Data Kedalaman	Data Kekeruhan	Data TDS	Data Aplikasi	Data Firebase	Data Microcontroller
1	7,9	24,3 °C	211,2 CM	10 NTU	220 PPM	Sesuai	Sesuai	Sesuai
2	8,25	23 °C	272,8 CM	10 NTU	198 PPM	Sesuai	Sesuai	Sesuai
3	7,5	24,6 °C	250,8 CM	0,1 NTU	245 PPM	Sesuai	Sesuai	Sesuai
4	7,63	25,7 °C	240,9 CM	0,1 NTU	139 PPM	Sesuai	Sesuai	Sesuai
5	8,21	24,9 °C	294,8 CM	100 NTU	300 PPM	Sesuai	Sesuai	Sesuai
6	8,96	26,31 °C	294,8 CM	100 NTU	329 PPM	Sesuai	Sesuai	Sesuai
7	8,14	25,43 °C	272,8 CM	100 NTU	311 PPM	Sesuai	Sesuai	Sesuai
8	8,58	26,21 °C	240,9 CM	50 NTU	283 PPM	Sesuai	Sesuai	Sesuai
9	7,16	24,91 °C	294,80 CM	1 NTU	101 PPM	Sesuai	Sesuai	Sesuai
10	8,9	28,1 °C	240,8 CM	200 NTU	398 PPM	Sesuai	Sesuai	Sesuai

Setelah melakukan pengujian dengan hasil seperti pada Tabel 6.12, dari 10 kali pengujian mendapatkan hasil dari data pada *microcontroller* lalu ketika data diterima oleh firebase dan data ketika data diterima oleh aplikasi android kesemuanya sesuai tanpa ada data yang hilang atau rusak. Dengan pengujian ini didapatkan kesimpulan dalam proses pengiriman data dari purwarupa alat menuju aplikasi android dengan perantara firebase tidak terdapat data yang hilang. Dengan hasil pengujian ini dapat menjadi referensi untuk penelitian selanjutnya atau penelitian yang ingin menggunakan firebase untuk menjadi *database*. Berikut contoh pengujian ini seperti pada Gambar 6.18 di atas ini. Gambar 6.18 merupakan salah satu contoh pengujian ke 9 dengan data hasil akuisisi sensor berupa pH air : 7,16, suhu air : 24,81 °C, TDS : 101, kedalaman air : 294,80 centimeter dan kekeruhan air sebesar 1 NTU serta hasil klasifikasi k-NN nya BAGUS.



https://skripsi98-e3d0d.firebaseio.com/

skripsi98-e3d0d

Counter: "79"

Gerak: "1"

Hasil Monitoring: "BAGUS"

KedalamanAIR: "294.80"

KekeruhanAIR: "1.00"

PHAIR: "7.16"

SuhuAIR: "24.81"

TDSAIR: "101" x



Gambar 6.17 Pengujian *Reliabilitas* Proses Pengiriman Data Ke *Firestore Database*

## 6.6 Pengujian Keseluruhan Sistem Gerak Purwarupa Alat

Pengujian keseluruhan sistem gerak purwarupa alat merupakan sebuah rangkaian pengujian dengan melakukan pengamatan pergerakan purwarupa alat sesuai dengan perintah si pengguna melalui aplikasi android. Pengujian ini dilakukan sebanyak 10 kali dengan perintah pergerakan yang berbeda beda untuk mengetahui respons dari sistem berdasarkan perintah yang diterima.

### 6.6.1 Tujuan

Tujuan dari pengujian ini agar diketahui apakah proses pengiriman data dari aplikasi android hingga diterima *microcontroller* terjadi kendala atau tidak. Dengan melakukan pengujian ini si peneliti bisa mengetahui tingkat akurasi pengiriman data dari android menuju node MCU.

### 6.6.2 Prosedur

1. Menyalakan *driver* motor yang digunakan sebagai pemerai daya untuk *microcontroller* arduino nano, node MCU dan kelima sensor yang digunakan.
2. Menghubungkan node MCU dengan laptop untuk melihat data yang dikirim dari aplikasi android melalui serial monitor arduino IDE.
3. Membuka aplikasi android dan memberikan perintah kepada alat berupa pergerakan seperti "MAJU, MUNDUR, KANAN, KIRI, STOP" sebanyak sepuluh kali.
4. Membuka firebase melalui google chrome untuk mengetahui data yang dikirim dari aplikasi android.
5. Melihat data yang dikirim aplikasi android melalui firebase ke node MCU melalui serial monitor arduino IDE.



### 6.6.3 Hasil Pengujian dan Analisis

Setelah melakukan pengujian sesuai dengan prosedur di atas, maka akan didapatkan hasil pengujian sesuai dengan tujuan atau tidak. Pengujian ini dengan cara mengamati proses pengiriman data yang berupa perintah dari aplikasi android untuk menggerakkan purwarupa alat. Dengan pengujian ini diharapkan si peneliti dapat mengetahui apakah terjadi data hilang dalam proses pengiriman data dari aplikasi hingga ke purwarupa alat dengan perantara firebase. Hasil pengujian dapat dilihat seperti pada Tabel 6.13.

**Tabel 6.13 Tabel Hasil Pengujian Keseluruhan Sistem Gerak Purwarupa Alat**

Pengujian ke	Perintah Aplikasi	Data FIREBASE	Data yang diterima Microcontroller	Validasi
1	MAJU	1	1	Sesuai
2	MUNDUR	2	2	Sesuai
3	KIRI	3	3	Sesuai
4	KANAN	4	4	Sesuai
5	STOP	0	0	Sesuai
6	KIRI	3	3	Sesuai
7	MAJU	1	1	Sesuai
8	MUNDUR	2	2	Sesuai
9	KANAN	4	4	Sesuai
10	STOP	0	0	Sesuai

Hasil pengujian sebanyak 10 kali sesuai dengan Tabel 6.13 di atas, semua pengujian yang dilakukan semuanya sesuai dengan perintah aplikasi android. Dengan hasil tersebut dapat diketahui bahwa proses pengiriman data dari aplikasi android untuk menggerakkan purwarupa alat sesuai hingga perintah tersebut diterima oleh purwarupa alat yang dibuat. Berikut contoh pengujian keseluruhan gerak purwarupa alat seperti pada Gambar 6.13. Data yang dikirim aplikasi untuk pergerakan purwarupa alat berupa angka seperti pada Gambar 6.19 yaitu data 1 untuk perintah MAJU. Hal tersebut untuk mempermudah dalam proses inialisasi data dalam proses pengiriman data.

```

HASIL : BAGUS
pH : 7.00
Suhu : 24.00
Kedalaman : 298.00
Kekeruhan : 5.00
TDS : 200.00
Gerak Kapal : MAJU
WAKTU : 3.6200

skripsi98-e3d0d
Counter: "210"
Gerak: "1"
Hasil Monitoring: "BAGUS"
KedalamanAIR: "298.00"
KekeruhanAIR: "5.00"
PHAIR: "7"
SuhuAIR: "24.00"
TDSAIR: "200"
    
```

**Gambar 6.18 Gambar Pengujian Keseluruhan Sistem Gerak Purwarupa Alat**



## BAB 7 KESIMPULAN

Pada akhir penelitian terdapat kesimpulan yang didapatkan dari keseluruhan hasil pengujian yang sesuai prosedur. Selain terdapat kesimpulan, pada bab ini terdapat saran yang diberikan oleh peneliti untuk menjadi referensi penelitian selanjutnya menjadi lebih baik dan akurat.

### 7.1 Kesimpulan

Setelah melakukan pengujian maka akan mendapatkan kesimpulan dengan acuan rumusan masalah dan hasil pengujian. Kesimpulan pada penelitian ini sebagai berikut ini :

1. Sistem untuk klasifikasi dan *monitoring* kualitas air bendungan dengan menggunakan metode k-NN dapat dibangun menggunakan 2 *microcontroller* yaitu arduino nano dan node MCU. Arduino nano digunakan untuk melakukan akuisisi data dari 5 sensor yang terdiri dari sensor TDS, sensor pH, sensor kedalaman air, sensor kekeruhan air dan sensor suhu air. Node MCU digunakan untuk melakukan proses klasifikasi k-NN dengan data dari arduino nano dan melakukan proses mengirim data menuju firebase serta melakukan proses menerima data dari firebase. Data yang dikirim berupa data dari 5 sensor, data hasil klasifikasi k-NN dan data yang diterima node MCU dari firebase berupa data untuk menggerakkan 2 motor DC sehingga purwarupa alat bisa bergerak sesuai dengan perintah.
2. Sistem pada penelitian ini menggunakan 5 sensor yang terdiri dari sensor pH, sensor suhu air, sensor kedalaman air, sensor kekeruhan air, sensor TDS. Untuk sensor pH setelah dilakukan pengujian didapatkan rata-rata *error* sebesar 7,97 %. Sensor suhu memiliki akurasi *error* sebesar 1,86 %. Akurasi sensor kedalaman air adalah 2,06 % serta sensor TDS yang digunakan mendapatkan akurasi *error* sebesar 5,86 %. Untuk sensor kekeruhan si peneliti tidak bisa mendapatkan akurasi *error* pada sensor karena tidak ada nya alat pembanding untuk parameter kekeruhan air. Setelah melakukan pengujian maka akan didapatkan hasil akurasi *error* dari 5 sensor yang digunakan pada penelitian ini.
3. Pengujian dengan metode k-NN untuk menentukan kualitas air yang telah dilakukan mendapatkan akurasi sebesar 92%. Pengujian ini menggunakan data uji sejumlah 25 data yang berbeda dan dilakukan sebanyak 25 kali pengujian. Terdapat 2 data pada proses pengujian k-NN yang tidak sesuai dengan kelas data yang asli. Pengujian klasifikasi k-NN ini menggunakan nilai  $K = 5$  untuk hasil yang terbaik dan terdapat 23 data pengujian yang sesuai dengan data kelas yang asli.
4. Jumlah waktu komputasi sistem berisi waktu komputasi arduino nano untuk akuisisi data dari 5 sensor ditambah waktu komputasi node MCU untuk klasifikasi k-NN dan proses mengirim data ke firebase. Proses pengujian ini dilakukan sebanyak 10 kali dengan waktu komputasi rata-rata pada keseluruhan sistem sebesar 4135 *ms*. Untuk waktu komputasi arduino nano memiliki rata-rata 1017 *ms* dan untuk waktu komputasi node MCU rata-rata sebesar 3391 *ms*. Untuk



penggunaan *memory* pada arduino nano sebesar 13870 *bytes* dari 30720 *bytes* dan 463 *bytes* dari 2048 *bytes* untuk *global variabel* sedangkan untuk node MCU penggunaan *memory* nya sebesar 36884 *bytes* dari 1044464 *bytes* dan 30716 *bytes* dari 81920 *bytes* untuk *global variabel*.

5. Pengujian tingkat *reliabilitas* sistem dalam proses mengirim dan mengambil data pada firebase dilakukan sebanyak 10 dengan cara melihat data yang di kirim oleh sistem ke firebase, melihat data yang ada pada firebase dan melihat data yang ditampilkan pada aplikasi *smartphone* apakah ada data yang hilang atau tidak. Pengujian tersebut didapatkan hasil bahwa sistem yang dibuat dapat mengirim data pada firebase dan dapat menampilkan data tersebut pada aplikasi android serta sistem dapat menggerakkan purwarupa alat sesuai dengan perintah dari aplikasi pengguna.

## 7.2 Saran

Saran dari si peneliti untuk menjadi referensi jika melakukan pengembangan dari penelitian ini atau penelitian yang berhubungan dengan kualitas air adalah sebagai berikut ini :

1. Menggunakan 1 *microcontroller* yang memenuhi kebutuhan dari sistem yaitu memiliki modul WiFi dan memiliki pin Analog lebih dari 3 sehingga dapat mempersingkat waktu komputasi jika menggunakan 1 *microcontroller* saja.
2. Menambah parameter kualitas air yang di teliti untuk meningkatkan hasil klasifikasi kualitas air tersebut
3. Menggunakan metode klasifikasi yang lain untuk mengetahui akurasi dari metode klasifikasi yang lain.
4. Melakukan proses paralel pada proses pengiriman data menuju *database* untuk mempercepat waktu komputasi dari sistem dan menggunakan *database* yang lain untuk mengetahui fungsi *database* yang lainnya.

**DAFTAR REFERENSI**

- A, Andang. et al., 2019. Investigation Of Ultrasonik Sensor Type Jsn-Sr04t Performance As Flood Elevation Detection.
- Arnomo Rio Adi., 2018 Implementasi Algoritma *K-Nearest Neighbor* Untuk Identifikasi Kualitas Air (Studi Kasus: Pdam Kota Surakarta). S1 STMIK Sinar Nusantara Surakarta 6(1),p. 1.
- Arduino.cc, 2018. WHAT IS ARDUINO?. [Online] Available at : <https://www.arduino.cc/en/Guide/Introduction> [Diakses 08 Desember 2019]
- Arduino, 2019. Arduino. [Online] Available at : <https://www.arduino.cc/en/Guide/ArduinoNano> [Diakses 12 Desember 2019]
- Cholilulloh Muchammad., 2018. Implementasi Metode *Fuzzy* Pada Kualitas Air Kolam Bibit Lele Berdasarkan Suhu Dan Kekeruhan 2(5), pp. 1813-1822.
- Codepolitan.com, 2017. Mengapa Harus Belajar Android Dengan Android Studio. [Online] Available at : <https://www.codepolitan.com/mengapa-harus-belajar-android-dengan-android-studio-59bfc3146686f> [Diakses 21 September 2019]
- Effendi, Hefni., 2003. TELAHAH KUALITAS AIR. Yogyakarta : Kanisius.
- Firebase, 2019. FIREBASE REALTIME DATABASE [Online] Available at : <https://firebase.google.com/docs/database/> [Diakses 12 Oktober 2019]
- G Muthuraman, S. Sasakila. 2014. Removal Of Turbidity Form Drinking Water Using Natural Coagulants 20(4), pp. 1727-1731.
- Han J and Kamber M. 2006. *DATA MINING : CONCEPT AND TECHNIQUES*. NEW YORK: MORGAN KAUFMANN PUBLISHER.
- Ichsan, M. H. H., Kurniawan, W., & Huda, M. 2016. *Water Quality Monitoring with Fuzzy Logic Control Based on Graphical Programming*. *Telkomnika*, 14(4), 1446-1453.
- Mahyudin., 2015. Analisis Kualitas Air Dan Strategi Pengendalian Pencemaran Air Sungai Metro Di Kota Kepanjen Kabupaten Malang 6(2), pp. 105-114.
- Maulana, R., Kurniawan, W., & Fahmi, H. Z. (2018). *Noise Reduction on the Tilt Sensor for the Humanoid Robot Balancing System Using Complementary Filter*. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (JTIK)*, 1(11), pp.1376-1384.
- Mutrofin, Siti. et al., 2014 Optimization Techniques Modified K-Nearest Neighbor Classification Using Genetic Algorithm 10(1), pp. 130-134.
- Mahida, U.N. 1993. Pencemaran Air Dan Pemanfaatan Limbah Industri. Edisi Keempat. Jakarta.: PT. Rajawali Grafindo



Peraturan Pemerintah Nomor 20 tahun 1990 tentang Pengendalian Pencemaran Air

Peraturan Pemerintah Nomor 416 tahun 1990 tentang Syarat-syarat Dan Pengawasan Kualitas Air.

Peraturan Pemerintah Nomor 82 tahun 2001 tentang Pengelolaan Kualitas Air Dan Pengendalian Pencemaran Air.

Peraturan Pemerintah Nomor 37 tahun 2011 tentang Bendungan.

Permana, Yudi Yulius, 2015 *Monitoring Kualitas Air Secara Real-Time Terintegrasi Integrated Real-Time Water Quality Monitoring*, Bandung: Pusat Penelitian Elektronika dan Telekomunikasi, Lembaga Ilmu Pengetahuan Indonesia.

Rangkuti Syahban, 2016, *Arduino Dan Proteus Simulasi Dan Praktik*, Bandung, Informatika.

Rosada, A., Ichsan, M. H. H., & Setyawan, G. E. (2019). Sistem Irigasi Pada Sawah Bertingkat Menggunakan Wireless Sensor Network. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (JPTIIK)* e-ISSN, 3(4), pp.3971-397

Saputra Akip., 2016. Pengukur Kadar Keasaman Dan Kekeruhan Air Berbasis Arduino.

Sugianti, Y., M. R. A. Putri dan Krismono. 2015. Karakteristik Komunitas Dan Kelimpahan Fitoplankton Di Danau Talaga, Sulawesi Tengah. *Limnotek*. 22(1), pp. 86 - 95 .

Wu X, Kumar V., 2009. *THE TOP TEN ALGORITHMS IN DATA MINING*. NEW YORK: CRC PRESS.

Yolanda, S. D., Maulana, R., & Ichsan, M. H. H. (2018). Scoring System Otomatis Pada Lomba Menembak Dengan Target Silhouette Hewan Menggunakan Metode K-Nearest Neighbor (K-NN). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (JPTIIK)*, 3(1), pp, 23 - 32.



## LAMPIRAN A – DATA LATIH

Data Ke-	pH	Suhu	Kekeruhan	Kedalaman	TDS	Klasifikasi Kelas
1	6,11	25,13 °C	100 NTU	211,2 CM	410 PPM	Jelek
2	5,9	26,95 °C	100 NTU	198 CM	530 PPM	Jelek
3	9,8	30,89 °C	200 NTU	176 CM	498 PPM	Jelek
4	9,14	29,52 °C	200 NTU	176 CM	450 PPM	Jelek
5	5,47	26,95 °C	100 NTU	198 CM	624 PPM	Jelek
6	5,11	29,7 °C	100 NTU	180,4 CM	581 PPM	Jelek
7	6,94	30 °C	200 NTU	240,84 CM	561 PPM	Jelek
8	5,86	28,78 °C	100 NTU	211,4 CM	599 PPM	Jelek
9	6,17	28,53 °C	200 NTU	198 CM	482 PPM	Jelek
10	8,76	29,91 °C	500 NTU	286 CM	714 PPM	Jelek
11	6,15	22 °C	50 NTU	272,8 CM	298 PPM	Sedang
12	6,78	21,98 °C	50 NTU	299,2 CM	315 PPM	Sedang
13	8,79	23,5 °C	50 NTU	286 CM	330 PPM	Sedang
14	8,9	22,81 °C	50 NTU	281,6 CM	328 PPM	Sedang
15	7,81	24,15 °C	100 NTU	220 CM	410 PPM	Sedang
16	8,61	26,98 °C	50 NTU	250,8 CM	325 PPM	Sedang
17	7,43	25,8 °C	50 NTU	308 CM	310 PPM	Sedang
18	9,1	27,5 °C	100 NTU	250,8 CM	405 PPM	Sedang
19	6,5	22,11 °C	50 NTU	286 CM	215 PPM	Sedang
20	8,8	26,7 °C	50 NTU	286 CM	250 PPM	Sedang
21	8,62	26,79 °C	50 NTU	294,9 CM	315 PPM	Sedang
22	8,21	26,81 °C	50 NTU	299,2 CM	299 PPM	Sedang
23	8,95	27,39 °C	100 NTU	211,2 CM	310 PPM	Sedang
24	8,76	27,1 °C	50 NTU	220 CM	325 PPM	Sedang
25	8,88	27,64 °C	50 NTU	250,8 CM	340 PPM	Sedang
26	8,91	26,95 °C	50 NTU	228,8 CM	310 PPM	Sedang
27	6,23	21,98 °C	100 NTU	272,8 CM	380 PPM	Sedang
28	7,9	24,81 °C	50 NTU	286 CM	340 PPM	Sedang



29	8,54	25,14 °C	100 NTU	303,6 CM	290 PPM	Sedang
30	8,46	25,6 °C	100 NTU	286 CM	284 PPM	Sedang
31	7,47	24,81 °C	10 NTU	228,8 CM	100 PPM	Bagus
32	7,78	25 °C	10 NTU	211,2 CM	198 PPM	Bagus
33	8,46	25,68 °C	10 NTU	250,8 CM	227 PPM	Bagus
34	8,37	23,5 °C	10 NTU	299,2 CM	132 PPM	Bagus
35	8,13	23,25 °C	10 NTU	220 CM	154 PPM	Bagus
36	7,81	25,1 °C	10 NTU	294,8 CM	210 PPM	Bagus
37	7,61	25,68 °C	10 NTU	211,2 CM	94 PPM	Bagus
38	7,72	25,9 °C	10 NTU	220 CM	190 PPM	Bagus
39	7,39	24,81 °C	10 NTU	294,8 CM	212 PPM	Bagus
40	7,43	24,15 °C	10 NTU	299,12 CM	184 PPM	Bagus
41	7,4	24,5 °C	10 NTU	299,12 CM	179 PPM	Bagus
42	7,1	26,1 °C	10 NTU	330 CM	198 PPM	Bagus
43	8,27	26,45 °C	10 NTU	325,6 CM	241 PPM	Bagus
44	7,81	26,9 °C	10 NTU	308 CM	205 PPM	Bagus
45	8,19	25,1 °C	10 NTU	308 CM	261 PPM	Bagus
46	8,05	25,46 °C	10 NTU	272,8 CM	236 PPM	Bagus
47	8	26 °C	0,01 NTU	211,2 CM	280 PPM	Bagus
48	7,05	24,5 °C	10 NTU	250,8 CM	215 PPM	Bagus
49	7,1	25,1 °C	0,1 NTU	211,1 CM	123 PPM	Bagus
50	7,91	25,6 °C	0,1 NTU	228,8 CM	251 PPM	Bagus



## LAMPIRAN B – DATA UJI

Pengujian Ke-	pH	Suhu	Kekeruhan	Kedalaman	TDS	Klasifikasi Kelas
1	8,45	25,06 °C	0,1 NTU	250,8 CM	91 PPM	Bagus
2	7,9	24,3 °C	1 NTU	211,2 CM	220 PPM	Bagus
3	7,1	25,2 °C	0,1 NTU	220,8 CM	180 PPM	Bagus
4	8,15	26 °C	0,1 NTU	272,8 CM	150 PPM	Bagus
5	7,5	24,6 °C	0,1 NTU	250,8 CM	245 PPM	Bagus
6	7,63	25,7 °C	0,1 NTU	240,9 CM	139 PPM	Bagus
7	7,82	24,95 °C	0,1 NTU	294,8 CM	100 PPM	Bagus
8	7,16	24,81 °C	10 NTU	294,8 CM	101 PPM	Bagus
9	8,25	23 °C	10 NTU	272,8 CM	198 PPM	Bagus
10	7,18	23,43 °C	0,1 NTU	211,2 CM	99 PPM	Bagus
11	8,41	25,88 °C	50 NTU	272,8 CM	300 PPM	Sedang
12	8,21	24,9 °C	100 NTU	294,8 CM	315 PPM	Sedang
13	8,8	26,19 °C	50 NTU	211,2 CM	340 PPM	Sedang
14	8,69	25,13 °C	100 NTU	220 CM	354 PPM	Sedang
15	8,14	25,43 °C	100 NTU	272,8 CM	311 PPM	Sedang
16	8,58	26,21 °C	50 NTU	240,9 CM	283 PPM	Sedang
17	7,91	24,5 °C	50 NTU	211,2 CM	256 PPM	Sedang
18	7,84	24,13 °C	50 NTU	240,9 CM	241 PPM	Sedang
19	8,96	26,31 °C	100 NTU	294,8 CM	329 PPM	Sedang
20	8,58	25,89 °C	100 NTU	294,8 CM	350 PPM	Sedang
21	5,99	26,4 °C	200 NTU	272,8 CM	430 PPM	Jelek
22	8,9	28,1 °C	200 NTU	240,8 CM	398 PPM	Jelek
23	6,2	27,5 °C	100 NTU	176 CM	453 PPM	Jelek
24	6,5	28,9 °C	100 NTU	211,2 CM	517 PPM	Jelek
25	7	28,36 °C	200 NTU	211,2 CM	381 PPM	Jelek



## LAMPIRAN C – KODE PROGRAM ARDUINO NANO

No	Baris Program
1	float waktu;
2	int conter = 0;
3	float Data2[5];
4	
5	void setup() {
6	Serial.begin(9600);
7	setOUTPUT1();
8	setbegin();
9	TDSbegin();
10	}
11	
12	void loop() {
13	pH_Air();
14	Suhu_Air();
15	Kedalaman_Air();
16	Kekeruhan_Air();
17	TDS_Air();
18	
19	String payload = "";
20	payload += "\\ph\":";
21	payload += Data2[0];
22	payload += "\\suhu\":";
23	payload += Data2[1];
24	payload += "\\kedalaman\":";
25	payload += Data2[2];
26	payload += "\\kekeruhan\":";
27	payload += Data2[3];
28	payload += "\\TDS\":";
29	payload += Data2[4];
30	payload += "\\counter\":";
31	payload += conter;
32	payload += "};
33	char attributes [200];
34	payload.toCharArray( attributes, 200 );
35	Serial.println(attributes);
36	conter++;
37	delay(500);
38	}
39	



## LAMPIRAN D – KODE PROGRAM NODE MCU

No	Baris Program
1	#include <ESP8266WiFi.h>
2	#include <ArduinoJson.h>
3	#include <FirebaseArduino.h>
4	
5	//#define WIFI_SSID "WIFIGRATIS"
6	//#define WIFI_PASSWORD "123456789"
7	#define WIFI_SSID "Kontrakan Atom"
8	#define WIFI_PASSWORD "mabestayo007"
9	#define FIREBASE_HOST "skripsi-e3d0d.firebaseio.com"
10	#define FIREBASE_AUTH "4XD8gHxqNZDP3Cluqbkwgsrftasdac"
11	
12	#define MOTOR1A D8 //8
13	#define MOTOR1B D6 //6
14	
15	#define MOTOR2A D5 //5
16	#define MOTOR2B D2 //2
17	
18	String kontrol;
19	const byte numChars = 200;
20	char receivedChars[numChars]; // an array to store the
21	received data
22	boolean newData = false;
23	
24	String input;
25	void recvWithEndMarker();
26	void showNewData();
27	
28	WiFiClient wclient;
29	
30	unsigned long start, finish;
31	float waktu;
32	int COUNTER = 0;
33	float Data[5]={12,23,21,22,22};
34	float Data2[5];
35	float DataNormalisasi[5];
36	float jarak[50][2];
37	
38	int DataNormalisasiINT[5];
39	float DataNormalisasiFinal[5];
40	
41	void setup() {
42	Serial.begin(9600);
43	
44	pinMode(MOTOR1A, OUTPUT);
45	pinMode(MOTOR1B, OUTPUT);
46	pinMode(MOTOR2A, OUTPUT);
47	pinMode(MOTOR2B, OUTPUT);
48	
49	// connect to wifi.
50	WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
51	Serial.print("connecting");
52	while (WiFi.status() != WL_CONNECTED) {
53	Serial.print(".");
54	delay(500);



```

55     }
56     Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
57     Serial.println("Siap Bosqu");
58 }
59
60 void loop() {
61     start = millis();
62     recvWithEndMarker();
63     showNewData();
64
65     StaticJsonBuffer<300> jsonBuffer;
66     JsonObject & root = jsonBuffer.parseObject(input);
67
68     // Test if parsing succeeds.
69     if (!root.success()) {
70         Serial.println("parseObject() failed");
71         return;
72     }
73
74     String pHAir = root["ph"];
75     String SuhuAir = root["suhu"];
76     String kedalamanAir = root["kedalaman"];
77     String kekeruhanAir = root["kekeruhan"];
78     String TDSAir = root["TDS"];
79     String Hasil = root["Hasil"];
80     String Counter = root["counter"];
81
82     Data2[0] = pHAir.toFloat();
83     Data2[1] = SuhuAir.toFloat();
84     Data2[2] = kedalamanAir.toFloat();
85     Data2[3] = kekeruhanAir.toFloat();
86     Data2[4] = TDSAir.toFloat();
87
88     DataNormalisasi[0] = (Data2[0]-0)/14; //0-14
89     DataNormalisasi[1] = (Data2[1]-0)/85; //0-100
90     DataNormalisasi[2] = (Data2[2]-20)/450; //20-1011
91     DataNormalisasi[3] = (Data2[3]-0)/1000; //0-3000
92     DataNormalisasi[4] = (Data2[4]-0)/1000; //0-1000
93
94     DataNormalisasiINT[0] = round(DataNormalisasi[0] * 1000);
95     DataNormalisasiFinal[0] = (float)DataNormalisasiINT[0] /
96     1000;
97     DataNormalisasiINT[1] = round(DataNormalisasi[1] * 1000);
98     DataNormalisasiFinal[1] = (float)DataNormalisasiINT[1] /
99     1000;
100    DataNormalisasiINT[2] = round(DataNormalisasi[2] * 1000);
101    DataNormalisasiFinal[2] = (float)DataNormalisasiINT[2] /
102    1000;
103    DataNormalisasiINT[3] = round(DataNormalisasi[3] * 1000);
104    DataNormalisasiFinal[3] = (float)DataNormalisasiINT[3] /
105    1000;
106    DataNormalisasiINT[4] = round(DataNormalisasi[4] * 1000);
107    DataNormalisasiFinal[4] = (float)DataNormalisasiINT[4] /
108    1000;
109
110    //Serial.println("-----");
111    //Serial.println("Hasil Perhitungan Euclidean");
112    int DataJarakHitung = 0;
113    int label = 1;

```



```

114     for(int a=0;a<50;a++){
115         jarak[a][0] = 0; //reset data
116         for(int b=0;b<5;b++){
117             jarak[a][0] += pow( Datalatih (a, b) -
118             DataNormalisasiFinal[b],2); //data uji - data
119             sensor(ph,kedalaman,kejernihan,suhu,tds)
120         }
121         jarak[a][0] = sqrt(jarak[a][0]); //akar
122         switch(a){
123             case 0...9:
124                 jarak[a][1] = 0; //angka 0 untuk label jelek
125                 break;
126             case 10...29:
127                 jarak[a][1] = 1; //angka 1 untuk label sedang
128                 break;
129             case 31...49:
130                 jarak[a][1] = 2; //angka 2 untuk label bersih
131                 break;
132         }
133     }
134     // Serial.println("-----");
135     // SORTING DATA
136     for(int i=0;i<50;i++){
137         for(int j=i+1;j<50;j++){
138
139             if(jarak[i][DataJarakHitung]>jarak[j][DataJarakHitung]){
140                 float temp = jarak[i][DataJarakHitung];
141                 jarak[i][DataJarakHitung] = jarak[j][DataJarakHitung];
142                 jarak[j][DataJarakHitung] = temp;
143
144                 temp = jarak[i][label];
145                 jarak[i][label] = jarak[j][label];
146                 jarak[j][label] = temp;
147             }
148         }
149     }
150     float satu = 0,dua = 0,tiga = 0;
151     String hasil;
152     Serial.println("-----");
153     Serial.println("K = 5");
154     for(int a=0;a<5;a++){
155         Serial.print("nilai");
156         Serial.print(a);
157         Serial.print(" ");
158         Serial.print(jarak[a][DataJarakHitung]);
159         Serial.print(" ");
160         Serial.print(jarak[a][label]);
161         Serial.println();
162     }
163     Serial.println("Perhitungan label");
164     if(jarak[a][label] == 0.00)
165     {
166         satu += 1/ pow(jarak[a][DataJarakHitung],2);
167     }
168     else if(jarak[a][label] == 1.00)
169     {
170         dua += 1/ pow(jarak[a][DataJarakHitung],2);
171     }
172     }

```



```

182     else if(jarak[a][label] == 2.00)
183     {
184         tiga += 1/ pow(jarak[a][DataJarakHitung],2);
185     }
186     }
187     Serial.println("-----");
188     jelek indeks label ke 0, sedang label ke 1 dan bagus label
189     ke 2
190     Serial.println("Nilai Indeks ");
191     Serial.print("Nilai JELEK :");
192     Serial.println(satu, 7);
193     Serial.print("Nilai SEDANG :");
194     Serial.println(dua, 7);
195     Serial.print("Nilai BAGUS :");
196     Serial.println(tiga, 7);
197     if(satu > dua && satu > tiga){
198         hasil = "JELEK";
199     }
200     else if(dua > satu && dua > tiga){
201         hasil = "SEDANG";
202     }
203     else if(tiga > dua && tiga > satu){
204         hasil = "BAGUS";
205     }
206     Serial.println("-----");
207     Serial.print("HASIL :");
208     Serial.println(hasil);
209     Firebase.setString("PHAIR", pHAir);
210     Firebase.setString("SuhuAIR", SuhuAir);
211     Firebase.setString("KedalamanAIR", kedalamanAir);
212     Firebase.setString("KekeruhanAIR", kekeruhanAir);
213     Firebase.setString("TDSAIR", TDSAir);
214     Firebase.setString("Hasil Monitoring", hasil);
215     Firebase.setString("Counter", Counter);
216     Serial.print("pH : ");
217     Serial.println(pHAir);
218     Serial.print("Suhu : ");
219     Serial.println(SuhuAir);
220     Serial.print("Kedalaman : ");
221     Serial.println(kedalamanAir);
222     Serial.print("Kekeruhan : ");
223     Serial.println(kekeruhanAir);
224     Serial.print("TDS : ");
225     Serial.println(TDSAir);
226     Serial.print("Gerak Kapal :");
227     Firebase.getString("Gerak");
228     String Gerak1 = Firebase.getString("Gerak");
229     Serial.println(Gerak1);
230     if(Gerak1 == "0"){ //STOP
231         digitalWrite(MOTOR1A, LOW);
232         digitalWrite(MOTOR1B, LOW);
233         digitalWrite(MOTOR2A, LOW);
234         digitalWrite(MOTOR2B, LOW);
235         Serial.println("STOP");
236     }else if(Gerak1 == "1"){ //MAJU
237         digitalWrite(MOTOR1A, LOW);
238         digitalWrite(MOTOR1B, HIGH);

```



```

231 digitalWrite(MOTOR2A, LOW);
232 digitalWrite(MOTOR2B, HIGH);
233 Serial.println("MAJU");
234 }else if(Gerak1 == "2"){ //MUNDUR
235 digitalWrite(MOTOR1A, HIGH);
236 digitalWrite(MOTOR1B, LOW);
237 digitalWrite(MOTOR2A, HIGH);
238 digitalWrite(MOTOR2B, LOW);
239 Serial.println("MUNDUR");
240 }else if(Gerak1 == "3"){ //KIRI
241 digitalWrite(MOTOR1A, LOW);
242 digitalWrite(MOTOR1B, HIGH);
243 digitalWrite(MOTOR2A, HIGH);
244 digitalWrite(MOTOR2B, LOW);
245 Serial.println("KIRI");
246 }else if(Gerak1 == "4"){ //KANAN
247 digitalWrite(MOTOR1A, HIGH);
248 digitalWrite(MOTOR1B, LOW);
249 digitalWrite(MOTOR2A, LOW);
250 digitalWrite(MOTOR2B, HIGH);
251 Serial.println("KANAN");
252 }
253 finish=millis();
254 waktu = (finish - start)/1000.0;
255 Serial.print("WAKTU : ");
256 Serial.println(waktu,4);
257 Serial.println("-----");
258 delay(100);
259 }
260 void recvWithEndMarker() {
261 static byte ndx = 0;
262 char endMarker = '\n';
263 char rc;
264 while (Serial.available() > 0 && newData == false) {
265 rc = Serial.read();
266 if (rc != endMarker) {
267 receivedChars[ndx] = rc;
268 ndx++;
269 if (ndx >= numChars) {
270 ndx = numChars - 1;
271 }
272 } else {
273 receivedChars[ndx] = '\0'; // terminate the string
274 ndx = 0;
275 newData = true;
276 }
277 }
278 }
279 void showNewData() {
280 if (newData == true) {
281 // Serial.print("This just in...");
282 Serial.println(receivedChars);
283 input = receivedChars;
284 newData = false;
285 }
286 }

```



## LAMPIRAN E – PP 20 TAHUN 1990 TENTANG PENGENDALIAN PENCEMARAN AIR

### 1. DAFTAR KRITERIA KUALITAS AIR GOLONGAN A

NO.	PARAMETER	SATUAN	KADAR MAKSIMUM	KETERANGAN
<ul style="list-style-type: none"> <li>• FISIKA</li> </ul>				
1.	Bau	-	-	Tidak berbau
2.	Jumlah zat padat terlarut (TDS)	mg/L	1.000	
3.	Kekeruhan	Skala NTU	5	
4.	Rasa	-	-	Tidak berasa
5.	Suhu	°C	Suhu udara ± 3° C	
6.	Warna	Skala TCU	15	
<ul style="list-style-type: none"> <li>• KIMIA</li> <li>a. KIMIA ANORGANIK</li> </ul>				
1.	Air raksa	mg/L	0.001	
2.	Alumunium	mg/L	0.2	
3.	Arsen	mg/L	0.05	
4.	Barium	mg/L	1.0	
5.	Besi	mg/L	0.3	
6.	Fluorida	mg/L	0.5	
7.	Kadmium	mg/L	0.005	
8.	Kesadahan CaCO <sub>3</sub>	mg/L	500	
9.	Klorida	mg/L	250	
10.	Kromium, valensi 6	mg/L	0.05	
11.	Mangan	mg/L	0.1	
12.	Natrium	mg/L	200	
13.	Nitrat, sebagai N	mg/L	10	
14.	Nitrit, sebagai N	mg/L	1.0	
15.	Perak	mg/L	0.05	
16.	pH	-	6.5 – 8.5	Merupakan batas minimum

### 2. DAFTAR KRITERIA KUALITAS AIR GOLONGAN B

NO.	PARAMETER	SATUAN	KADAR MAKSIMUM	KETERANGAN
<ul style="list-style-type: none"> <li>• FISIKA</li> </ul>				
1.	Suhu	°C	Suhu air normal	
2.	Zat padat terlarut	mg/L	1.000	
<ul style="list-style-type: none"> <li>• KIMIA</li> <li>a. KIMIA ANORGANIK</li> </ul>				
1.	Air raksa	mg/L	0.001	
2.	Amoniak bebas	mg/L	0.5	
3.	Arsen	mg/L	0.05	
4.	Barium	mg/L	1.0	
5.	Besi	mg/L	5.0	
6.	Fluorida	mg/L	1.5	
7.	Kadmium	mg/L	0.01	
8.	Klorida	mg/L	600	
9.	Kromium, valensi 6	mg/L	0.05	
10.	Mangan	mg/L	0.5	
11.	Nitrat, sebagai N	mg/L	10	
12.	Nitrit, sebagai N	mg/L	1.0	
13.	Oksigen terlarut (DO)	mg/L	*	* Air permukaan dianjurkan lebih besar atau sama dengan 6
14.	pH	-	5 – 9	
15.	Selenium	mg/L	0.01	
16.	Seng	mg/L	5	
17.	Sianida	mg/L	0.1	
18.	Sulfat	mg/L	400	
19.	Sulfida, sebagai H <sub>2</sub> S	mg/L	0.1	
20.	Tembaga	mg/L	1.0	
21.	Timbal	mg/L	0.1	

**3. DAFTAR KRITERIA KUALITAS AIR GOLONGAN C**

NO.	PARAMETER	SATUAN	KADAR MAKSIMUM	KETERANGAN
•	FISIKA			
1.	Suhu	°C	Suhu air normal ± 3° C	
2.	Zat padat terlarut	mg/L	1.000	
•	KIMIA			
a.	KIMIA ANORGANIK			
1.	Air raksa	mg/L	0.002	
2.	Amoniak bebas	mg/L	0.02	
3.	Arsen	mg/L	1.0	
4.	Fluorida	mg/L	1.5	
5.	Kadmium	mg/L	0.01	
6.	Klorin bebas	mg/L	0.003	
7.	Kromium, valensi 6	mg/L	0.05	
8.	Nitrit, sebagai N	mg/L	0.06	
9.	Oksigen terlarut (DO)	mg/L	*	* Disyaratkan lebih besar dari 3
10.	pH	-	6 - 9	
11.	Selenium	mg/L	0.05	
12.	Seng	mg/L	0.02	
13.	Sianida	mg/L	0.02	
14.	Sulfida, sebagai H <sub>2</sub> S	mg/L	0.002	
15.	Tembaga	mg/L	0.02	
16.	Timbal	mg/L	0.03	

**4. DAFTAR KRITERIA KUALITAS AIR GOLONGAN D**

NO.	PARAMETER	SATUAN	KADAR MAKSIMUM	KETERANGAN
•	FISIKA			
1.	Daya hantar listrik	umhos/cm	2.250	Tergantung dengan jenis tanaman. Kadar maksimum tersebut untuk tanaman yang tidak peka.
2.	Suhu	°C	Suhu air normal	Sesuai dengan kondisi setempat.
3.	Zat padat terlarut	mg/L	2.000	Tergantung dengan jenis tanaman. Kadar maksimum tersebut untuk tanaman yang tidak peka.
•	KIMIA			
a.	KIMIA ANORGANIK			
1.	Air raksa	mg/L	0.005	
2.	Arsen	mg/L	1.0	
3.	Boron	mg/L	1.0	
4.	Kadmium	mg/L	0.01	
5.	Kobalt	mg/L	0.2	
6.	Kromium, valensi 6	mg/L	1	
7.	Mangan	mg/L	2.0	
8.	Na (garam alkali)	mg/L	60.0	
9.	Nikel	mg/L	0.5	