

Monitoring Kualitas Air Tambak Udang Dengan Metode Data Logging dan Algoritma KNN Berbasis Internet Of Things

Harry Pratama Ramadhan^{1*)}, Condro Kartiko², Agi Prasetiadi³

¹Jurusan S1 Teknik Informatika, Fakultas Informatika, Institut Teknologi Telkom, Purwokerto

²Jurusan S1 Rekayasa Perangkat Lunak, Fakultas Informatika, Institut Teknologi Telkom, Purwokerto

³Jurusan S1 Teknik Informatika, Fakultas Informatika, Institut Teknologi Telkom, Purwokerto

^{1,2,3} Jl. DI Panjaitan No.128, Pancurawis, Purwokerto Kidul, Kec. Purwokerto Sel., Kabupaten Banyumas, Jawa Tengah
email: ¹me.harrypratama@gmail.com, ²condro.kartiko@ittelkom-pwt.ac.id, ³agi@ittelkom-pwt.ac.id

Abstract – Based on the experience of researchers, many shrimp farming entrepreneurs in Indonesia have gone bankrupt, this is due to the large costs for one laboratory check so that shrimp farming experiences harvest failure because many shrimps get disease and die. In this study, a water quality monitoring tool was made from vannamei shrimp ponds using methods data logging based on water temperature values and the algorithm k-nearest neighbors to predict shrimp health and shrimp pond water conditions from water movement. The device data logger uses a NodeMCU ESP8266 microcontroller, LDR sensor, and a Dallas DS18B20 water temperature sensor, then the logger data is connected to Google's database service, the Firebase realtime database to store water quality monitoring data. There are web services that are hosted on website Heroku's to run the algorithm k-nearest neighbors using the command http request from an Android application built using the framework Flutter. The Flutter Android application contains monitoring widgets, detailed monitoring and predictions with an accuracy rate of 99.9% for shrimp health and 97.5% for water conditions.

Keywords – Data Logging, Flutter Android, Internet of Things, K-Nearest Neighbors, Monitoring, Water Quality.

Abstrak – Berdasarkan pengalaman peneliti, banyak pengusaha budi daya tambak udang di Indonesia mengalami bangkrut, hal tersebut disebabkan karena biaya yang besar untuk satu kali cek laboratorium sehingga budi daya tambak udang mengalami gagal panen dikarenakan banyak udang yang terserang penyakit dan mati. Pada penelitian ini dibuat sebuah alat pemantauan kualitas air dari tambak udang vannamei menggunakan metode data logging berdasarkan nilai suhu air dan algoritma k-nearest neighbors untuk memprediksi kesehatan udang dan kondisi air tambak udang dari pergerakan air. Perangkat data logger menggunakan mikrokontroler NodeMCU ESP8266, sensor LDR, dan sensor suhu air Dallas DS18B20, kemudian data logger terhubung ke layanan basis data milik Google yaitu Firebase realtime database untuk menyimpan data pemantauan kualitas air. Terdapat layanan web services yang di hosting pada layanan website milik Heroku untuk menjalankan algoritma k-nearest neighbors menggunakan perintah httprequest dari aplikasi Android yang dibangun menggunakan framework Flutter. Aplikasi Flutter Android berisi widget monitoring, detail monitoring dan prediksi dengan tingkat akurasi yang diberikan sebesar 99.9 % untuk kesehatan udang serta 97.5 % untuk kondisi air.

Kata Kunci – Data Logging, Flutter Android, Internet of Things, K-Nearest Neighbors, Kualitas Air, Monitoring.

***) penulis korespondensi:** Harry Pratama Ramadhan
Email: me.harrypratama@gmail.com

I. PENDAHULUAN

Budi daya tambak merupakan peluang usaha yang berpotensi untuk mendapatkan keuntungan yang sangat besar. Berdasarkan jumlah ekspor komoditas tambak udang milik Kementerian Kelautan dan Perikanan Republik Indonesia dari tahun 2015 hingga 2019 khususnya untuk komoditas udang memiliki volume ekspor sebesar 62,64 ribu ton dengan penjualan sebesar 514,32 juta US Dollar. Walaupun angka tersebut merupakan penurunan dari tahun sebelumnya yaitu pada tahun 2018 komoditas udang memiliki volume ekspor sebesar 64,98 ribu ton dengan penjualan sebesar 602,81 juta US Dollar, komoditas udang masih menempati posisi pertama sebagai komoditas yang penjualan serta volume nya terbesar dari komoditas lainnya. Gbr 1 merupakan tren nilai dan volume ekspor komoditas udang.^[1]

TREN EKSPOR HASIL PERIKANAN PER KOMODITAS 2015-2019
Periode Januari-April 2015-2019



Gbr. 1 Tren Nilai dan Volume Ekspor Komoditas Udang Tahun 2015 – 2019
Direktorat Jenderal Penguatan Daya Saing Produk Kelautan Dan Perikanan [1]

Namun, masih banyak pengusaha budi daya tambak udang yang mengalami kerugian dikarenakan komoditas udang merupakan komoditas yang mudah terserang penyakit. Hal tersebut menyebabkan kerugian yang cukup besar karena udang mengalami gagal panen. Adapun virus yang kerap

menyerang dimasa pertumbuhan udang dan mengurangi ketahanan hidup udang hingga mencapai 30 % yaitu *white feses disease* (WFD) dan *white spot viruses* (WSSV).^[2] Penyakit tersebut menyerang udang apabila kualitas air tambak udang kurang baik, adapun parameter yang digunakan untuk mengukur kualitas air yaitu berdasarkan nilai pH, suhu, kecerahan, dan organisme.^[3]

A. Rumusan Masalah

Rumusan masalah pada penelitian berikut ini berdasarkan latar belakang permasalahan di atas yaitu sebagai berikut :

Bagaimana tingkat keberhasilan implementasi metode *data logging* dan algoritma *k-nearest neighbors* pada penelitian ini dapat melakukan pemantauan kualitas air dengan baik, sehingga dapat meningkatkan produksi komoditas udang di Indonesia ?

B. Tujuan

Adapun tujuan dari adanya penelitian berikut ini yaitu sebagai berikut :

Membuat sistem pemantauan untuk menganalisis kualitas air tambak udang secara *real-time* dengan metode *data logging* menggunakan NodeMCU yang berperan sebagai *data logger* serta algoritma *k-nearest neighbors* dapat memberikan nilai prediksi dengan akurasi yang baik berdasarkan kondisi kualitas air dan kesehatan udang. Diharapkan dengan adanya alat monitoring kualitas air tersebut, hasil produksi komoditas udang selanjutnya semakin meningkat.

C. Batasan Masalah

Adapun batasan masalah dari adanya penelitian berikut ini yaitu sebagai berikut :

1. Sistem *monitoring* hanya dapat dilakukan pada siang hari, karena sensor LDR memanfaatkan sinar matahari.
2. Sistem ini hanya dapat berjalan ketika cuaca tidak hujan, karena perangkat data logger merupakan perangkat elektronik yang mudah rusak apabila terkena air.
3. Sistem deteksi pergerakan air sangat tergantung pada tingkat kekeruhan air, karena tingkat kekeruhan air memiliki nilai refraksi cahaya matahari yang berbeda.

II. PENELITIAN YANG TERKAIT

Terdapat beberapa penelitian yang terkait dengan penelitian monitoring kualitas air tambak udang, berikut ini diantaranya yaitu penulisan terkait desain *monitoring* kualitas air tambak udang berbasis *internet of things* dan pasar udang secara *real time* untuk mendukung industri 4.0 dijelaskan bahwa implementasi sistem *monitoring* tersebut menggunakan beberapa komponen seperti NodeMCU ESP8266, sensor, *firebase cloud storage*, aplikasi dan *website i-Tambak Monitoring* yang terkoneksi dengan *i-Tambak Marketplace* menerapkan metode pengembangan *software* yaitu metode *prototype* yang bertujuan untuk membantu industri 4.0 pasar udang dengan membuat pasar udang secara online serta aplikasi *monitoring* tambak udang untuk

membantu pengelola tambak udang mendapatkan informasi mengenai kualitas air tambak udang.^[4] Lalu untuk penelitian selanjutnya yaitu mengenai *monitoring* kualitas air untuk tambak udang berbasis *wireless sensor network* (WSN) dijelaskan bahwa parameter yang digunakan untuk pemantauan kualitas air pada penelitian berikut ini yaitu menggunakan tiga buah sensor yaitu sensor pH, sensor suhu dan sensor kekeruhan air, kemudian menggunakan Arduino untuk mengumpulkan data dan menggunakan Xbee untuk mengirim data ke perangkat penerima informasi berupa tampilan LCD. ESP8266 digunakan untuk menghubungkan perangkat ke jaringan internet yaitu data yang dikumpulkan akan disimpan pada basis data. Hasil dari penelitian kedua yaitu *wireless sensor network* dapat bekerja dengan baik dengan akurasi kalibrasi masing-masing sensor yaitu 97.76% untuk sensor suhu, 98.85% untuk sensor pH dan 99.73% untuk sensor kekeruhan.^[5]

Penelitian selanjutnya tentang *deep-learning* berbasis pendekatan untuk prakiraan kualitas air tambak udang secara intensif dijelaskan bahwa model *deep belief networks* (DBN) digunakan untuk meramalkan kondisi kualitas air dari tambak udang. Kualitas air diperkirakan menggunakan *Canadian water quality index* (WQI) yang diperoleh dari lapisan model simulasi. Parameter yang digunakan pada proses prakiraan kualitas air yaitu parameter suhu, oksigen terlarut, pH, salinitas, amonia, nitrat, nitrit, *orthophosphate* dan *chlorophyll*.. Hasil dari penelitian ini yaitu DBN memiliki potensi besar dalam untuk memprediksi kualitas air dan kemampuan akurasi nya terpenuhi dengan jumlah 10 lapisan tersembunyi RMSE. Performa dari DBN menjadi yang terbaik Ketika *learning rate* sebesar 0.01.^[6] Kemudian terdapat model prediksi kualitas air tambak udang dari penelitian lain yaitu menggunakan *adaptive neuro fuzzy inference system* (ANFIS), pada penelitian berikut ini menggunakan *mini computer raspberry pi* dan memiliki aplikasi pengendali kualitas air berbasis Android. ANFIS digunakan untuk mengendalikan beberapa parameter kualitas air yaitu suhu, keseimbangan pH, level ketinggian air dan oksigen terlarut. *Output* dari penggunaan ANFIS yaitu aktivasi pompa air untuk mencegah ketidakseimbangan air tambak dan memiliki mode otomatis dan manual. Hasil pengujian dari penelitian ini berupa pengujian kualitas air di Madura, Indonesia yang cuacanya sangat ekstrim, suhu air berkisar antara 35 hingga 40 derajat celcius yang tentunya sangat mempengaruhi kadar pH dan sistem ANFIS bekerja dengan sangat baik.^[7]

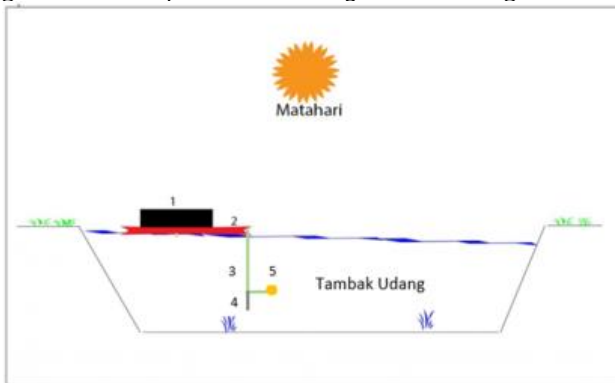
Sedangkan pada penelitian ini, yang akan disajikan yaitu tentang prediksi kualitas air tambak udang menggunakan mikrokontroler ESP8266 dengan parameter suhu dan kecerahan air, mendeteksi udang berdasarkan pergerakan air menggunakan algoritma *k-nearest neighbors* yang berfungsi untuk memberikan informasi mengenai kualitas air dan kesehatan udang berdasarkan data yang didapatkan dari sensor LDR dan memberikan informasi berupa suhu yang ideal untuk proses budi daya tambak udang dari data yang didapatkan dari sensor suhu air. Menurut Liao & Muarai (1986) keberhasilan dalam budidaya udang yaitu suhu air yang berkisar antara 20-30 °C.^[8]

III. METODE PENELITIAN

Metode yang digunakan dalam pengembangan sistem monitoring kualitas air tambak udang menggunakan metode SDCL *prototype*. Adapun beberapa tahapan dari SDLC *prototype* dimulai dari *analysis & initial requirements, design, implementation, evaluation, testing and maintenance*.

A. Analysis & Initial Requirements

Pada tahap ini dilakukan pengumpulan data yaitu berupa studi literatur dari jurnal sebelumnya yang dipaparkan pada Bab 2, observasi lokasi budi daya tambak, wawancara dengan pengelola tambak, kebutuhan pembangunan sistem monitoring kualitas air tambak dari segi *hardware* dan *software* dan setelah proses pembangunan alat selesai maka dilakukan pengambilan data sampel air tambak untuk pembuatan *dataset* algoritma *k-nearest neighbors*. Dari hasil wawancara dengan pengelola tambak, ditemukan bahwasannya *monitoring* kualitas air tambak berdasarkan pergerakan dan perubahan suhu air belum memiliki alat khusus. Pengelola dan pemilik tambak menginginkan alat *monitoring* kualitas air yang *portable*. Gbr 2 berikut ini merupakan rancangan dari perangkat *data logger* yang akan digunakan untuk proses *monitoring* tambak udang.



Gbr. 2 Rencana Bentuk Perangkat *Monitoring*

Berdasarkan rancangan di atas, terdapat beberapa penjelasan mengenai komponen apa saja yang digunakan pada Tabel I berikut ini :

TABEL I
KETERANGAN KOMPONEN PERANGKAT MONITORING

No	Nama	Keterangan
1	Black Box	Merupakan tempat dimana semua perangkat keras data logger terintegrasi menjadi satu
2	Pelampung	Untuk membantu perangkat keras data logger selalu di atas permukaan air (tidak tenggelam)
3	Pipa PVC	Tempat untuk meletakkan kabel sensor
4	Sensor Suhu	Sensor untuk mengukur suhu air
5	Sensor LDR	Sensor untuk mengukur pergerakan air dan udang di dalam air

Dalam proses pengembangan sistem *monitoring* kualitas air tambak udang, dibutuhkan beberapa *hardware* dan *software*. Tabel II berikut ini merupakan rincian *hardware* yang diperlukan dalam pembangunan sistem pada penelitian ini.

TABEL II
KEBUTUHAN *HARDWARE* PERANGKAT MONITORING

Nama	Spesifikasi	Jumlah
NodeMCU ESP8266	Versi 1.0 LoLin	1 Buah
	Wifi 802.11 b/g/n	
	Pin GPIO 13, ADC 1	
	RAM 64 KB	
	Flash Memory 4 MB	
Sensor LDR	Input Tegangan 5V	1 Buah
	Flying Fish – MH Sensor	
	4 Pin, VCC, GND, A0, D0	
Sensor Suhu	Input Tegangan 3V - 5V	1 Buah
	Dallas DS18B20	
	3 Pin, VCC, GND, Data	
Sensor DHT	Input Tegangan 3V – 5V	1 Buah
	DHT 11	
	3 Pin, VCC, GND, dan Data	
Kipas Pendingin	Input Tegangan 3V – 5V	1 Buah
	DC Brushless	
	Input Tegangan 12V	
Access Point	Ukuran 6x6 CM	1 Buah
	Tethering Dari Smartphone	
	4G LTE	
Mini UPS	Smart Portable Mini UPS	1 Buah
	Output Tegangan 12V	
	Kapasitas 2000 mAH	
Step Down	Step Down 12V to 5V	1 Buah
Black Box	Ukuran 22x15x7CM	1 Buah
Relay	Modul Relay 5V	1 Buah
	3 Pin, VCC, GND, dan Sig	
	1 Channel	
Kabel Jumper	Male to Female	1 Buah
	Male to Male	
	Female to Female	
Kabel Kecil	Kabel Pelangi	1 Set
Breadboard	Mini	1 Buah
Laptop	Windows 10 Pro	1 Buah
	Intel Pentium P6000	
	RAM 4 GB	
Smartphone	Android 10 (Q) 64 Bit	1 Buah
	Qualcomm Snapdragon 625	

Setelah kebutuhan *hardware* terpenuhi, agar perangkat *data logger* tidak tenggelam ketika berada di tambak, maka diperlukan beberapa komponen agar perangkat tersebut dapat mengapung. Tabel III berikut ini merupakan kebutuhan material pelampung perangkat monitoring.

TABEL III
KEBUTUHAN PELAMPUNG PERANGKAT MONITORING

Nama	Spesifikasi	Jumlah
Pelampung	Pelampung Renang	1 Buah
Pipa PVC	Ukuran $\frac{5}{8}$ Inch	1 Buah
T Dus	Ukuran $\frac{5}{8}$ Inch	3 Buah
Knee L	Ukuran $\frac{5}{8}$ Inch	10 Buah
Lem Pipa	Lem Pipa PVC	1 Buah

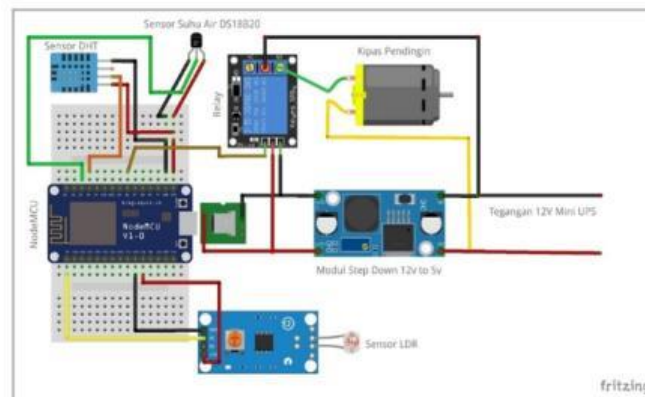
Agar sistem monitoring dapat bekerja dengan baik, maka diperlukan penulisan program untuk mikrokontroler dan aplikasi berbasis Android. Tabel IV berikut ini merupakan rincian kebutuhan *software* yang diperlukan.

TABEL IV
KEBUTUHAN SOFTWARE PERANGKAT MONITORING

Nama	Spesifikasi	Jumlah
Windows 10 Pro	Sistem Operasi	1 Buah
	Arsitektur 64 Bit	
	Versi 2004	
	Build 19041.388	
Visual Studio Code	Text Editor	1 Aplikasi
	Versi 1.47.2	
	Ekstensi Flutter	
Arduino IDE	Versi 1.8.12	1 Aplikasi
	Library :	
	StandardC++	
	FirebaseESP8266	
	SimpleTimer	
	DallasTemperature	
	OneWire	
	Arduino JSON	
	NTP Client	
Flutter SDK	Versi 1.17.5	1 Package
Firebase	Realtime Database	1 Akun
	User Authentication	
Heroku	Web hosting python	1 Akun
Python	Pyrebase versi 4	
	Sklearn.neighbors.	
	KNeighbors Classifier	
	Pandas	
	Numpy	
	Framework Flask	
Android Q	Security Patch 05-07-2020	

B. Design

Pada tahap ini, semua kebutuhan sistem yang sudah dirancang pada tahap sebelumnya dibentuk dalam skema dan *mock up* untuk perangkat *data logger* serta aplikasi Android. Gbr 3 berikut ini merupakan hasil perancangan skema komponen penyusun *data logger* dengan menggunakan aplikasi Fritzing.



Gbr. 3 Desain Skema Komponen *Data Logger*

Berdasarkan desain skema komponen *data logger* tersebut terdapat rincian hubungan antara pin komponen dengan mikrokontroler agar perangkat *data logger* dapat bekerja sebagaimana mestinya. Tabel V berikut ini merupakan tabel yang menjelaskan hubungan antara pin komponen dengan mikrokontroler maupun komponen lainnya.

TABEL V
HUBUNGAN ANTAR PIN KOMPONEN

Sensor / Module		PIN	
Nama	PIN	NodeMCU	PIN Eksternal
Sensor LDR	A0	A0	
	VCC	3.3 V	
	GND	G (Ground)	
Sensor DS18B20	VCC	3.3 V	
	Data	D2	
	GND	G (Ground)	
DHT 11	VCC	3.3 V	
	Data	D3	
	GND	G (Ground)	
Relay	VCC		Out + Step Down
	IN	D5	
	GND		Out – Step Down
	NC		Pin - Kipas DC
	C		Pin – Mini UPS

TABEL V
HUBUNGAN ANTAR PIN KOMPONEN (LANJUTAN)

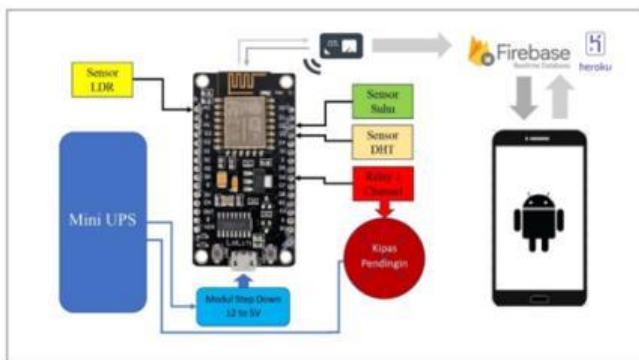
Sensor / Module		PIN	PIN
Nama	PIN	NodeMCU	Eksternal
Modul Step Down 12 V to 5 V	IN +		Pin + Mini UPS
	IN -		Pin - Mini UPS
	OUT +		Pin + Kabel USB
	OUT -		Pin - Kabel USB
Kipas DC	VCC		Pin + Relay
	GND		Pin - Mini UPS

Semua komponen tersebut akan disatukan kedalam desain alat yang terdapat pada Gbr 4 supaya alat tidak tenggelam ketika berada di kolam tambak.



Gbr. 4 Rancangan Perangkat Data Logger 3D

Secara keseluruhan, alur kerja desain sistem tersebut digambarkan melalui Gbr 5.



Gbr. 5 Desain Keseluruhan Sistem

Pengembangan sistem *monitoring* kualitas air tambak udang menggunakan Firebase *realtime database* untuk menyimpan data yang didapatkan oleh sensor suhu dan sensor LDR. Alasan menggunakan Firebase *realtime database* karena basis data tersebut cukup mudah untuk digunakan dan

mikrokontroler NodeMCU ESP8266 yang mendukung *library* dari Firebase seperti FirebaseESP8266 dan *library* JSON untuk input data ke basis data Firebase yaitu *library* ArduinoJSON. Gbr 6 berikut ini merupakan potongan kode yang digunakan untuk mengirim data sensor ke Firebase.

```
String Waktu = timeClient.getFormattedTime();
Firebase.setString(firebaseData, path + "/Waktu/Data", Waktu);
Firebase.pushString(firebaseData, datapath + "/Waktu", Waktu);
```

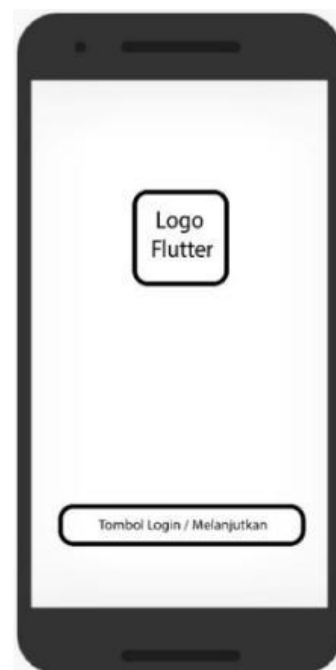
Gbr. 6 Kode Program Mengirim Data Ke Basis Data

Supaya mikrokontroler dan *web service* Heroku dapat mengakses Firebase *realtime database*, maka pada Gbr 7 dilakukan konfigurasi seperti berikut.

```
{
  "rules": {
    ".read": true,
    ".write": true,
    "Data Monitoring": {
      ".indexOn": ["Cerah", "CerahDeviasi", "CerahRataRata"]
    }
  }
}
```

Gbr. 7 Konfigurasi Rules Realtime Database

Sistem *monitoring* kualitas air tambak udang dilakukan dengan cara melihat informasi yang ditampilkan oleh aplikasi berbasis Android yang dibangun menggunakan *framework* Flutter yang ditunjukkan pada Gbr.8, Gbr.9, Gbr.10, Gbr. 11, Gbr. 12 dan Gbr. 13 berdasarkan desain *mock-up* aktivitas dari aplikasi *monitoring* dari tampilan *login* hingga info akun.



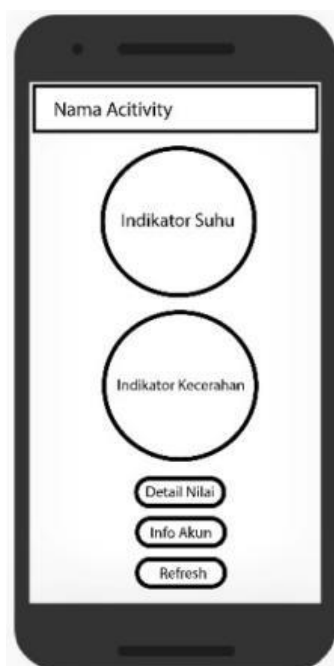
Gbr. 8 Desain Mock-up Tampilan Awal



Gbr. 9 Desain *Mock-up Login Akun Google*



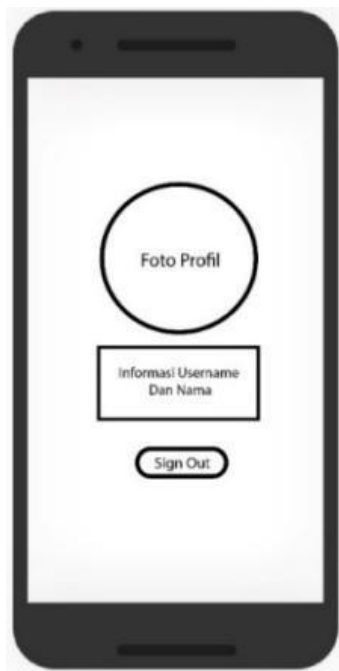
Gbr. 11 Desain *Mock-up Detail Monitoring*



Gbr. 10 Desain *Mock-up Monitoring*



Gbr. 12 Desain *Mock-up Prediksi KNN*



Gbr. 13 Desain Mock-up Info Akun

C. Implementation Tahap 1

Pada tahap ini, semua sistem yang sudah didesain pada tahap sebelumnya diimplementasikan menjadi implementasi *data logger*, implementasi basis data, implementasi algoritma dan perhitungan, implementasi antarmuka. Gbr 14 berikut ini merupakan hasil implementasi dari perangkat *data logger*.



Gbr. 14 Implementasi Komponen Perangkat Data Logger

Hasil dari perancangan Gbr.14 disatukan pada kerangka pelampung yang dibuat menggunakan pipa paralon $\frac{5}{8}$ inch sebagai kerangka nya pada Gbr. 15.



Gbr.15 Implementasi Final Perangkat Data Logger

Setelah implementasi perangkat *data logger* berhasil dilakukan, langkah selanjutnya yaitu implementasi perhitungan varian, standar deviasi dan rata-rata yang data nya akan digunakan untuk menentukan kualitas air tambak udang berdasarkan pergerakan air dengan bantuan algoritma *k-nearest neighbors* untuk menentukan kondisi air dan kesehatan udang. Rumus 1 berikut ini merupakan rumus konversi rentang nilai sensor LDR dari 0 sampai 1023 menjadi 0 sampai 100.

$$\text{convert} = \frac{\text{read_adc}(1) \times 100}{1023} \quad (1)$$

Keterangan :

convert = konversi nilai dari 0 sampai 1023 ke 0 sampai 100

read_adc = nilai sensor dari sebelum dikalibrasi

x100 = nilai sensor dari ADC dikalikan 100

1023 = hasil dari perkalian dibagi 1023

Tujuan dilakukan konversi nilai sensor LDR yaitu untuk memudahkan pembacaan nilai yang secara umum menggunakan rentang nilai 0 sampai 100. Sedangkan sensor suhu tidak dilakukan konversi karena sudah menunjukkan nilai suhu celcius. Untuk selanjutnya yaitu perhitungan varian dan standar deviasi. Rumus 2 berikut ini adalah rumus yang secara umum untuk menghitung varian dan Rumus 3 berikut ini adalah rumus yang digunakan untuk menghitung standar deviasi. Perbedaan antara varian dan standar deviasi yaitu standar deviasi merupakan turunan dari varian yaitu hasil akar dari nilai varian.

$$S^2 = \frac{n \sum_{i=1}^n X_i^2 - (\sum_{i=1}^n X_i)^2}{n(n-1)} \quad (2)$$

$$S = \sqrt{\frac{n \sum_{i=1}^n X_i^2 - (\sum_{i=1}^n X_i)^2}{n(n-1)}} \quad (3)$$

Keterangan :

S^2 = varian

S = standar deviasi

X_i = nilai x ke-i

$\bar{x}$ = rata-rata

n = ukuran sampel

Untuk rumus yang ketiga yaitu rumus perhitungan rata-rata. Rumus 4 berikut ini merupakan rumus rata-rata.

$$\text{mean} = \frac{\text{array_data}}{30} \quad (4)$$

Keterangan :

mean = rata - rata

array_data = data sensor yang ada di array

30 = data sensor pada array dibagi 30

Rumus di atas diterapkan ke dalam kode program dari Arduino IDE dan dilakukan upload ke mikrokontroler NodeMCU ESP8266. Kode program 1 berikut ini adalah kode program perhitungan varian, standar deviasi dan rata-rata dari sensor LDR dan sensor suhu

Kode Program 1 Perhitungan Varian, Standar Deviasi dan Rata-Rata Dari Sensor LDR dan Sensor Suhu

```
double s_arr[30] = {0,0,0,0,0 0,0,0,0,0,
                  0,0,0,0,0,
                  0,0,0,0,0,
                  0,0,0,0,0,
                  0,0,0,0,0};

double t_arr[30] = {0,0,0,0,0, 0,0,0,0,0,
                  0,0,0,0,0,
                  0,0,0,0,0,
                  0,0,0,0,0,
                  0,0,0,0,0};

int variance(double a[], double n)
{
    // Compute mean (average of elements)
    int sum = 0;
    for (int i = 0; i < n; i++)
        sum += a[i];
    double mean = (double)sum /
        (double)n;

    // Compute sum squared
    // differences with mean.
    double sqDiff = 0;
    for (int i = 0; i < n; i++)
        sqDiff += (a[i] - mean) *
            (a[i] - mean);
    return sqDiff / n;
}

double standardDeviation(double arr[], double n)
{
    return sqrt(variance(arr, n));
}

double mean(double arr[]){
    double avg = 0.0;
    double sum = 0.0;
    for (int i = 0; i < 30; i++)
    {
        sum += s_arr[i];
    }
    avg = sum / 30;
    return avg;
}

double meanT(double arr[]){
```

```
double avg = 0.0;
double sum = 0.0;
for (int i = 0; i < 30; i++)
{
    sum += t_arr[i];
}
avg = sum / 30;
return avg;
}

void geser_ke_kanan_suhu(){
    for (int i = 0; i <= 28; i++)
    {
        t_arr[30-i-1] = t_arr[30-i-2];
    }

    t_arr[0] = 0;
}

void geser_ke_kanan_cahaya(){
    for (int i = 0; i <= 28; i++)
    {
        s_arr[30-i-1] = s_arr[30-i-2];
    }

    s_arr[0] = 0;
}

void catat_suhu(double ini)
{
    geser_ke_kanan_suhu();
    t_arr[0] = ini;
}

void catat_cahaya(double ini)
{
    geser_ke_kanan_cahaya();
    s_arr[0] = ini;
}
```

Dari kode program di atas terdapat implementasi kode program dengan menggunakan array yang memiliki 30 elemen, tujuan dari implementasi kode program tersebut yaitu untuk menjalankan algoritma *shifting array*. Algoritma *shifting array* yaitu algoritma yang digunakan untuk menggeser kode program yang sudah ada sebelumnya ke elemen berikutnya ketika terdapat data baru yang masuk ke dalam *array* tersebut. Setelah data sensor berhasil diolah dengan implementasi rumus di atas, maka langkah selanjutnya yaitu mengirim data sensor ke basis data Firebase. Kode program 2 berikut ini merupakan kode program yang menjelaskan proses data sensor yang berhasil diolah oleh rumus di atas untuk selanjutnya dikirimkan ke Firebase *realtime database*.

Tabel VI berikut ini merupakan contoh implementasi dari pembuatan *dataset*.

TABEL VI
IMPLEMENTASI *DATASET*

D1	D2	D3	R1	R2	R3	K1	K2	K3	C
2.6 33	2.6 05	3.1 27	71. 715	71. 686	72. 060	76. 367	73. 632	81. 347	0
4.2 31	4.6 60	4.7 24	91. 155	90. 677	90. 182	81. 542	79. 882	82. 421	1
3.0 76	3.0 76	3.0 96	9.9 21	9.9 21	10. 042	7.9 10	8.6 91	12. 304	2
0.9 69	0.9 68	0.9 77	6.4 71	6.3 99	6.3 21	5.0 78	5.3 71	5.0 78	3
4.8 08	4.8 08	4.5 92	21. 236	21. 236	21. 041	24. 804	26. 953	23. 242	4

D1 sampai D3 = Data Standar Deviasi Kecerahan
R1 sampai R3 = Data Rata – Rata Kecerahan
K1 sampai K3 = Data Kecerahan
C = Kategori

```
[ ] X_train = XV[:,p_train,:l_train]
print(X_train)

C> [[ 2.63309729 2.60565915 3.1270356 ... 76.3671875 73.6328125
      81.34765625]
      [ 2.60565915 3.1270356 3.15277211 ... 73.6328125 81.34765625
      74.609375 ]
      [ 3.1270356 3.15277211 3.15273039 ... 81.34765625 74.609375
      72.265625 ]
      ...
      [ 0.80915745 0.81644014 0.78321888 ... 8.09140625 10.25390625
      8.3984375 ]
      [ 0.81644014 0.78321888 0.78519679 ... 10.25390625 8.3984375
      8.09140625 ]
      [ 0.78321888 0.78519679 0.82205985 ... 8.3984375 8.09140625
      10.7421875 ]]
```

Gbr. 16 Pembagian Data menjadi *Data Train* dan *Data Test*

Implementasi algoritma *k-nearest neighbors* pada python dijelaskan pada kode program 3 berikut ini.

```
def knnair():
    # Impor Dataset Kondisi Air
    dataset = pd.read_csv('dataset.csv')
    dataset_ = dataset.values
    XY = dataset[:,1:11]
    ukuran = XY.shape
```

```

p_train = int(ukuran[0]*0.8)
l_train = int(ukuran[1]-1)

# Split Dataset Kondisi Air Menjadi Training Set
dan Test Set
X_train = XY[:p_train,:l_train]
X_test = XY[257:,-1]
Y_train = XY[:p_train,-1:]
Y_test = XY[257:,-1:]

X_train, X_test, y_train, y_test = train_test_split(
    XY[:, :-1], XY[:, -1:], test_size=0.25)

# Fitting Simple K Nearest Neighbour ke Training
set
neigh = KNeighborsClassifier(n_neighbors=3)
neigh.fit(X_train, y_train)

# Memprediksi Data Keadaan Air Dengan KNN
Y_pred = neigh.predict(X_test)

A = []

CerahDeviasi =
db.child("Data
Monitoring/CerahDeviasi").order_by_key().
limit_to_last(3).get()
for CerahDeviasi in CerahDeviasi.each():
    A.append(CerahDeviasi.val())
    #print(CerahDeviasi.val())

CerahRataRata =
db.child("Data
Monitoring/CerahRataRata").order_by_key().
limit_to_last(3).get()
for CerahRataRata in CerahRataRata.each():
    A.append(CerahRataRata.val())
    #print(CerahRataRata.val())

Cerah =
db.child("Data Monitoring/Cerah").order_by_key().
limit_to_last(3).get()
for Cerah in Cerah.each():
    A.append(Cerah.val())
    #print(Cerah.val())

A = np.asarray(A)
A = A.reshape(1, 9)
hasil = neigh.predict(A)
hasilnya = str(hasil[0])

if ( hasilnya == "0.0" ):
    data = {"Data": "Kondisi Air 97.5 % Tenang /
Perangkat Tidak Di Air"}
    db.child("Monitoring/StatusAir").set(data)
elif ( hasilnya == "1.0" ):

```

```

data = {"Data": "Air 97.5 % Terlalu Keruh"}
    db.child("Monitoring/StatusAir").set(data)
elif ( hasilnya == "2.0" ):
    data = {"Data": "97.5 % Perangkat Tidak Di
Air"}
    db.child("Monitoring/StatusAir").set(data)
elif ( hasilnya == "3.0" ):
    data = {"Data": "97.5 % Air Tambak Tenang /
Perangkat Berada di Tepi"}
    db.child("Monitoring/StatusAir").set(data)
elif ( hasilnya == "4.0" ):
    data = {"Data": "97.5 % Arus Air Deras dan
Banyak Pergerakan"}
    db.child("Monitoring/StatusAir").set(data)

return hasilnya

```

Pada tahap *evaluation* terdapat perubahan pada cara pengolahan data suhu air yang sebelumnya akan diolah menggunakan algoritma *k-nearest neighbors* dirubah menjadi percabangan biasa pada kode program Arduino. Terdapat 4 status kondisi suhu air yang akan dikirimkan pada basis data dan diakses oleh aplikasi Android. Kode program 3 berikut ini merupakan percabangan dan pemberian status suhu air yang digunakan pada sistem *monitoring*.

Kode Program 3 Status Kondisi Suhu Air

```

String status1 = "Suhu Air Terlalu Dingin";
String status2 = "Suhu Air Ideal";
String status3 = "Suhu Air Hangat";
String status4 = "Suhu Air Panas";

void StatusSuhu()
{
    if (myDouble < 25 ){
        Firebase.setString(firebaseData,      path      +
"/StatusSuhu/Data",status1);
    }

    if (myDouble > 25 && myDouble < 27){
        Firebase.setString(firebaseData,      path      +
"/StatusSuhu/Data",status2);
    }

    if (myDouble > 27 && myDouble <= 30){
        Firebase.setString(firebaseData,      path      +
"/StatusSuhu/Data",status3);
    }
    if (myDouble > 30 && myDouble <= 35){
        Firebase.setString(firebaseData,      path      +
"/StatusSuhu/Data",status4);
    }
}

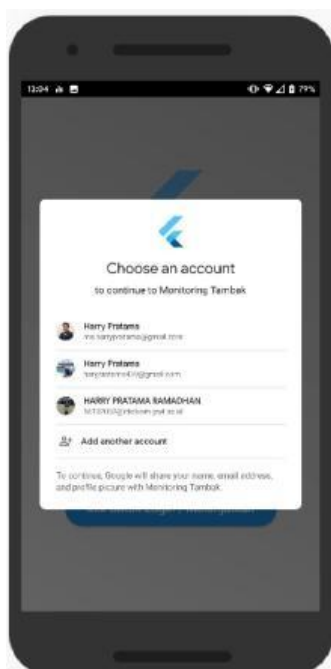
```

E. Implementation Tahap 2

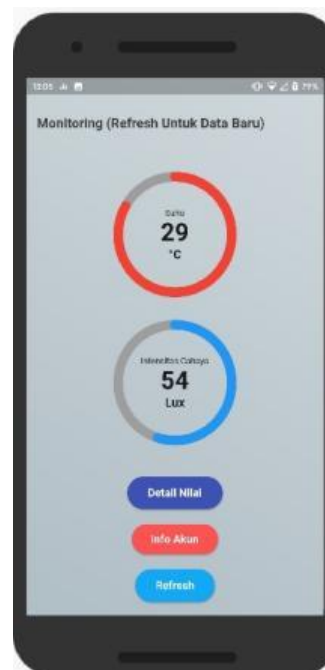
Setelah dilakukan beberapa evaluasi pada tahap sebelumnya, maka pembangunan aplikasi *monitoring* kualitas air dilakukan dengan beberapa tampilan yang sudah dijelaskan pada tahap *design*. Gbr.17, Gbr.18, Gbr.19, Gbr.20, Gbr.21 dan Gbr.22 merupakan hasil implementasi dari pembangunan aplikasi *monitoring*.



Gbr.17 Implementasi Tampilan Awal Aplikasi



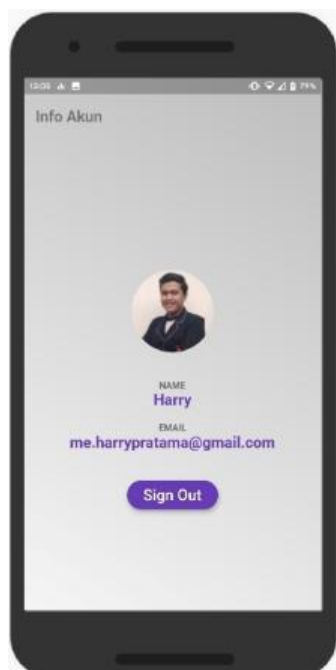
Gbr.18 Implementasi Tampilan Login



Gbr.19 Implementasi Tampilan Monitoring



Gbr.20 Implementasi Tampilan Detail



Gbr.21 Implementasi Info Akun

IV. HASIL DAN PEMBAHASAN

Bab ini akan membahas tentang beberapa pengujian dari implementasi sistem *monitoring* yang sudah dijelaskan pada bab metodologi penelitian. Pengujian pertama yaitu pengujian durabilitas sensor, tujuan dari pengujian ini untuk mengetahui kekuatan dari sensor LDR dan sensor suhu terhadap kondisi yang ekstrim. Tabel VII berikut ini merupakan hasil dari pengujian durabilitas sensor LDR dan Tabel VIII merupakan hasil pengujian durabilitas sensor suhu.

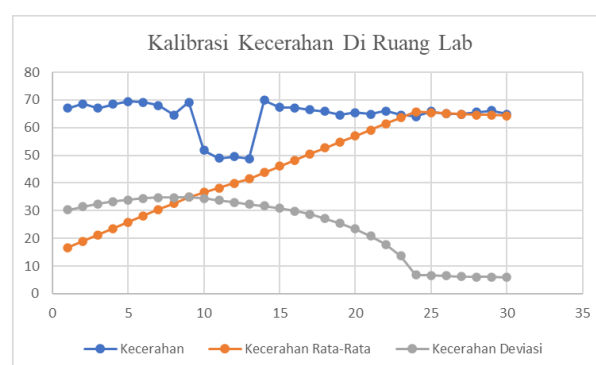
TABEL VII
HASIL PENGUJIAN DURABILITAS SENSOR LDR

Menit	Pengujian	Perubahan Nilai		Tercatat Pada Basis Data		Perubahan Nilai Pada Aplikasi	
		Y	T	Y	T	Y	T
3	Air Lab	✓		✓		✓	
	Kolam	✓		✓		✓	
	Tambak	✓		✓		✓	
6	Air Lab	✓		✓		✓	
	Kolam	✓		✓		✓	
	Tambak	✓		✓		✓	
9	Air Lab	✓		✓		✓	
	Kolam	✓		✓		✓	
	Tambak	✓		✓		✓	
12	Air Lab	✓		✓		✓	
	Kolam	✓		✓		✓	
	Tambak	✓		✓		✓	
15	Air Lab	✓		✓		✓	
	Kolam	✓		✓		✓	
	Tambak	✓		✓		✓	

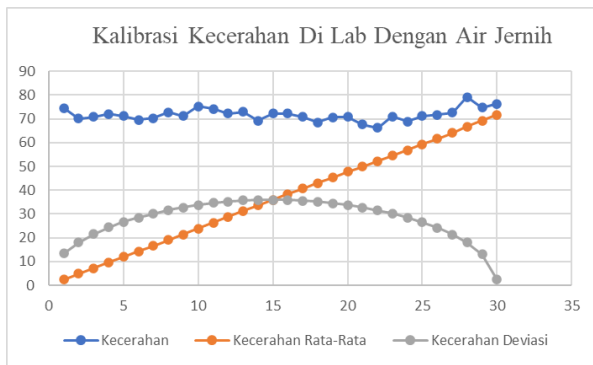
TABEL VIII
HASIL PENGUJIAN DURABILITAS SENSOR SUHU

Menit	Pengujian	Perubahan Nilai		Tercatat Pada Basis Data		Perubahan Nilai Pada Aplikasi	
		Y	T	Y	T	Y	T
3	Air Dingin	✓		✓		✓	
	Air Biasa	✓		✓		✓	
	Air Panas	✓		✓		✓	
6	Air Dingin	✓		✓		✓	
	Air Biasa	✓		✓		✓	
	Air Panas	✓		✓		✓	
9	Air Dingin	✓		✓		✓	
	Air Biasa	✓		✓		✓	
	Air Panas	✓		✓		✓	
12	Air Dingin	✓		✓		✓	
	Air Biasa	✓		✓		✓	
	Air Panas	✓		✓		✓	
15	Air Dingin	✓		✓		✓	
	Air Biasa	✓		✓		✓	
	Air Panas	✓		✓		✓	

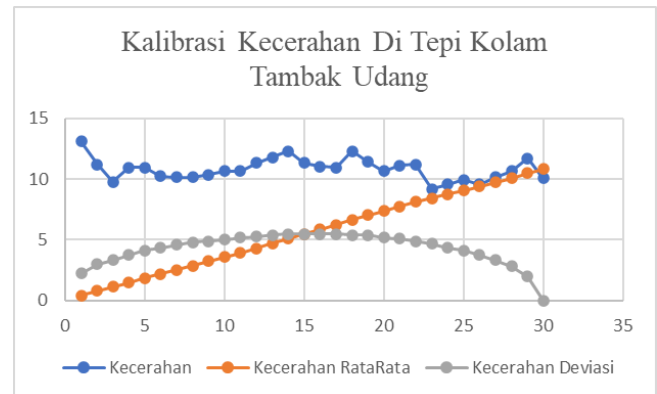
Setelah sensor diuji durabilitasnya, pengujian selanjutnya yaitu pengujian kalibrasi sensor LDR. Terdapat delapan scenario pengujian yang dilakukan untuk melakukan kalibrasi pada sensor. Pengujian kalibrasi yaitu menguji nilai sensor berdasarkan tiga puluh data pertama ketika perangkat *data logger* dihidupkan. Gbr.22, Gbr.23, Gbr.24, Gbr.25, Gbr.26, Gbr.27, Gbr.28 dan Gbr.29 merupakan grafik hasil pengujian kalibrasi sensor LDR berdasarkan skenario pengujiannya masing-masing.



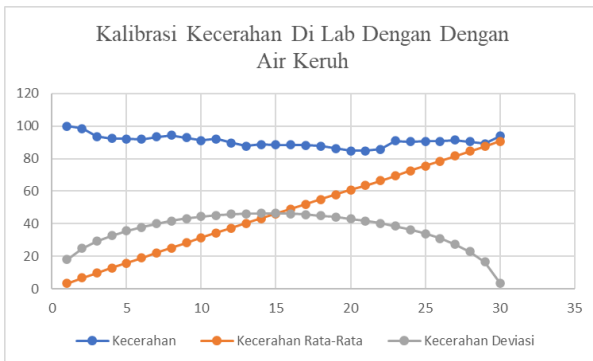
Gbr.22 Skenario Pengujian Kecerahan Di Ruang Lab



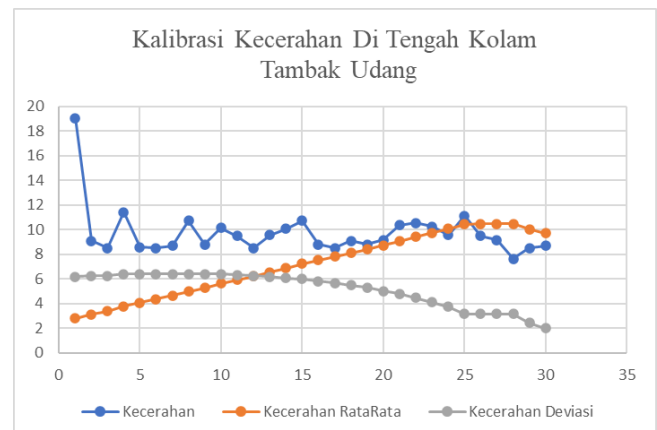
Gbr.23 Skenario Pengujian Kecerahan Air Jernih Lab



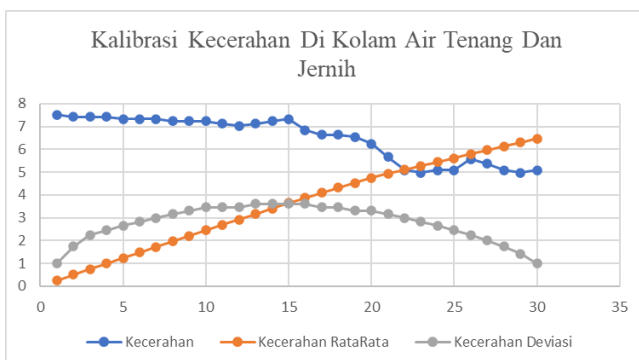
Gbr.27 Skenario Pengujian Kecerahan Di Tepi Kolam Tambak Udang



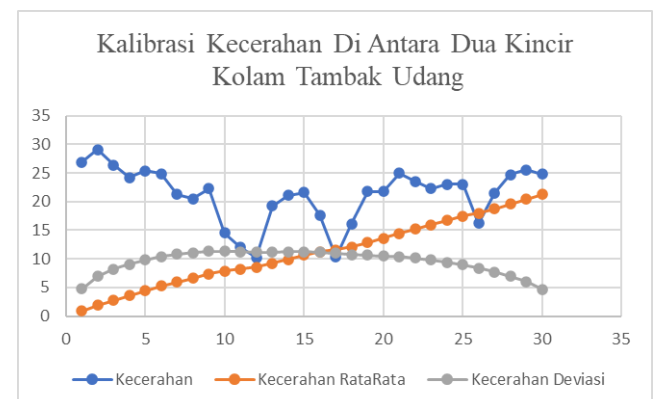
Gbr.24 Skenario Pengujian Kecerahan Air Keruh Lab



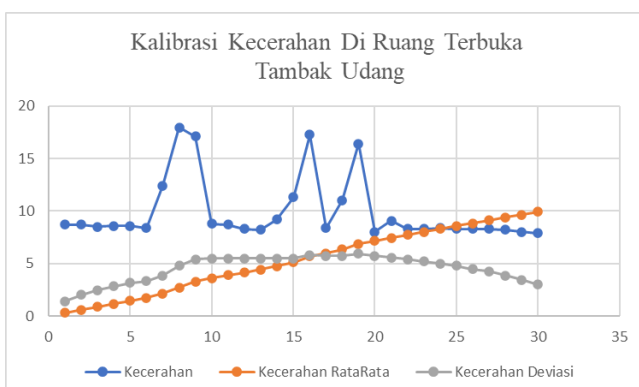
Gbr.28 Skenario Pengujian Kecerahan Di Tengah Kolam Tambak Udang



Gbr.25 Skenario Pengujian Kecerahan Air Tenang Kolam

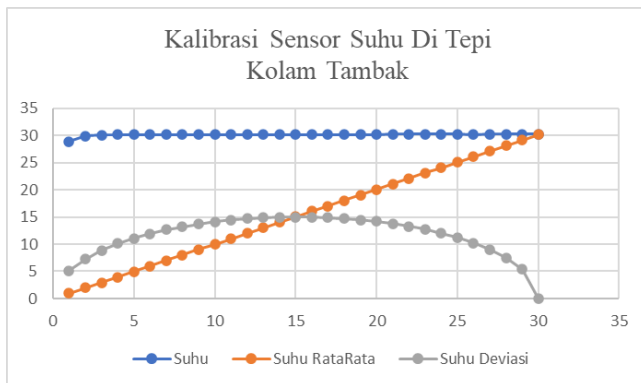


Gbr.29 Skenario Pengujian Kecerahan Di Antara Dua Kincir Kolam Tambak Udang

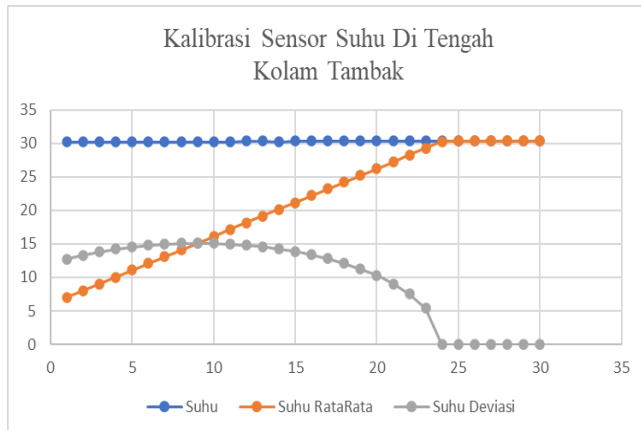


Gbr.26 Skenario Pengujian Kecerahan Ruang Terbuka Tambak

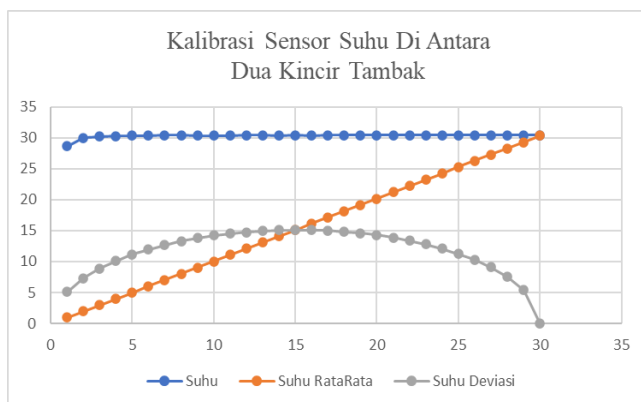
Selain pengujian kalibrasi sensor LDR, terdapat tiga skenario pengujian untuk sensor suhu air. Gbr.30, Gbr.31, dan Gbr.32 merupakan hasil pengujian kalibrasi sensor suhu air.



Gbr.30 Skenario Pengujian Suhu Air Di Tepi Kolam Tambak



Gbr.31 Skenario Pengujian Suhu Air Di Tengah Kolam Tambak



Gbr.32 Skenario Pengujian Suhu Air Di Antara Dua Kincir Kolam Tambak Udang

Setelah dilakukan pengujian kalibrasi, maka dari hasil tersebut didapatkan 40 data yang dijadikan untuk pembuatan *dataset* menggunakan algoritma *k-nearest neighbors*. *Dataset* tersebut dimasukkan kedalam program KNN yang dapat dilihat pada kode program 3 di atas. Untuk memastikan apakah nilai prediksi sudah memiliki tingkat akurasi yang baik, maka diperlukan adanya pengujian dari algoritma *k-nearest neighbors*. Gbr. 33 merupakan hasil dari pengujian *Y Pred Predict* pada algoritma *k-nearest neighbors*.

```
[ ] from sklearn.neighbors import KNeighborsClassifier
    neigh = KNeighborsClassifier(n_neighbors=3)
    neigh.fit(X_train, y_train)

[ ] Y_pred = neigh.predict(X_test)
    Y_pred

array([[2., 1., 3., 1., 3., 1., 0., 1., 3., 1., 0., 3., 1., 2., 4., 3., 2.,
        3., 3., 3., 3., 3., 3., 1., 1., 2., 3., 3., 4., 1., 2., 0., 0.,
        0., 4., 4., 1., 2., 1., 3., 1., 0., 1., 3., 1., 4., 1., 2., 2., 3.,
        1., 1., 4., 4., 4., 3., 3., 0., 0., 0., 4., 1., 1., 0., 3., 3.,
        3., 3., 3., 3., 0., 2., 3., 3., 3., 0., 4., 3.]])
```

Gbr.33 Pengujian Prediksi

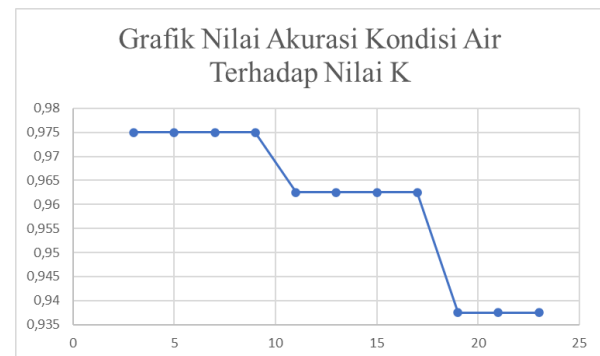
Selanjutnya, dilakukan pengujian akurasi data dengan menggunakan skor akurasi. Gbr.34 berikut ini merupakan nilai akurasi yang berhasil didapatkan algoritma *k-nearest neighbors*.

```
[ ] import sklearn
    sklearn.metrics.accuracy_score(y_test, Y_pred)

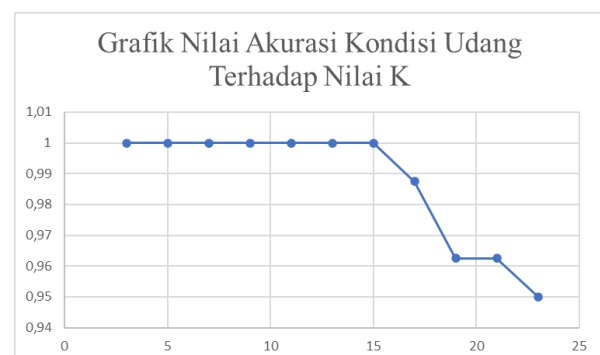
0.975
```

Gbr.34 Pengujian Akurasi Data

Untuk memastikan apakah akurasi tersebut merupakan akurasi terbaik, maka dilakukan pengujian akurasi dari $K = 3$ hingga $K = 25$. Gbr. 35 merupakan hasil pengujian akurasi kondisi air dan Gbr. 36 merupakan hasil pengujian akurasi kondisi kesehatan udang.



Gbr. 35 Grafik Pengujian Nilai K Kondisi Air

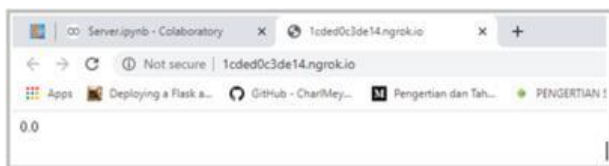


Gbr. 36 Grafik Pengujian Nilai K Kondisi Udang

Hasil kalibrasi menunjukkan bahwa semakin banyak nilai K, maka akurasi akan semakin menurun. Maka peneliti memutuskan untuk memilih $K = 3$ sebagai nilai K dari algoritma *k-nearest neighbors*. Selanjutnya yaitu pengujian *httprequest* algoritma *k-nearest neighbors*. Menggunakan *debug console* dari Visual Studio Code, peneliti akan menguji apakah *web server* berjalan dengan baik dengan *status code* 200. Gbr.37 berikut ini merupakan hasil pengujian dari *httprequest*.

Gbr. 37 Pengujian *httprequest*

Untuk hasil prediksi yang ditampilkan di *web service* hanya menunjukkan angka kategori seperti ada Gbr.38. Hasil prediksi akan mudah dipahami dengan diberi percabangan sebelum dikirim ke Firebase seperti pada Gbr.39 berikut ini.

Gbr.38 Pengujian *Web Services*

Gbr.39 Percabangan Hasil Prediksi

Hasil prediksi algoritma di atas akan dikirim ke basis data Firebase *realtime database* seperti yang ditunjukkan pada Gbr.40 berikut ini.



Gbr.40 Basis Data Prediksi

Untuk *dataset* menggunakan 40 data setelah tahap kalibrasi sensor. Data tersebut dibagi menjadi beberapa kategori yaitu 5 kategori untuk kondisi air dan 3 kategori

untuk kesehatan udang. Agar semakin memudahkan informasi yang didapatkan, maka dibuat tampilan antarmuka untuk melakukan prediksi pada aplikasi *monitoring* kualitas air. Tampilan aplikasi tersebut terdapat tombol prediksi untuk melakukan prediksi kondisi air dan kesehatan udang menggunakan algoritma *k-nearest neighbors*. Gbr.41 berikut ini merupakan antarmuka tambahan yaitu prediksi *k-nearest neighbors* pada aplikasi monitoring berbasis Android



Gbr.41 Antarmuka Prediksi KNN

V. KESIMPULAN

Dari hasil pengujian di atas dapat ditarik kesimpulan bahwa :

- Implementasi metode data logging untuk melakukan pemantauan dan pencatatan keadaan air dapat bekerja dengan baik, dimana untuk kedua sensor yaitu sensor suhu dan sensor LDR ketika dilakukan pengujian dapat mencatat data dengan baik dengan syarat koneksi internet yang digunakan harus stabil.
- Alat dapat untuk menentukan kualitas air dengan akurasi sebesar 97.5 % untuk kondisi air dan 99.9 % untuk kondisi kesehatan udang menggunakan program algoritma *k nearest neighbor* pada *web services* Heroku dapat berjalan dengan sesuai harapan dengan method *http request* dari aplikasi Android.
- Penelitian ini menghasilkan inovasi baru pada penentuan kualitas air, karena sebelumnya tambak yang dijadikan obyek penelitian belum memiliki alat penentu kualitas air berdasarkan pergerakan dan suhu air.

Terdapat beberapa kelemahan dari sistem ini yang masih perlu diperbaiki dan ditingkatkan kembali yaitu :

- Perhitungan nilai deviasi di bawah nol koma, ketika nol koma maka angka yang muncul seperti dibulatkan antara 1 atau 0
- Untuk saat ini NodeMCU ESP8266 masih melakukan pengiriman data ke Firebase *realtime database* ketika terkoneksi ke Internet. Koneksi internet yang tidak stabil sangat berpengaruh terhadap pengiriman data dari NodeMCU ESP8266 ke Firebase *realtime database*.

Adapun saran dari peneliti untuk pengembangan selanjutnya yaitu :

- Ditambahkan sensor lain seperti salinitas dan sensor pendeteksi pertumbuhan plankton.
- Dirancang sebuah alat yang memiliki sifat anti air.
- Ditambahkan metode *local save* agar dapat menangani permasalahan pengiriman data ketika koneksi internet bermasalah. Data disimpan ke penyimpanan tambahan misalnya memori NodeMCU ESP8266, kemudian data tersebut akan dikirimkan ke basis data apabila koneksi internet tersedia.

UCAPAN TERIMA KASIH

Terima kasih kepada penulis dua dan tiga selaku dosen pembimbing saya selama penelitian ini dan terima kasih kepada saudara sadu selaku pengelola tambak yang sudah bersedia untuk memberikan informasi mengenai budi daya tambak udang.

DAFTAR PUSTAKA

- [1] djpd1, "Trend Ekspor Impor Periode Januari - April Tahun 2015-2019," *Direktorat Jenderal Penguatan Daya Saing Produk Kelautan Dan Perikanan*, 2019. [Online]. Available: <https://knp.go.id/djpdspkp/artikel/12107-trend-ekspor-impor-periode-januari-april-tahun-2015-2019>. [Accessed Sep. 23, 2020].
- [2] S. Adiwijaya, "Indonesia Berpeluang Kembangkan Industri Akuakultur," *Tempo*, 2016. [Online]. Available: <https://bisnis.tempo.co/read/799904/indonesia-berpeluang-kembangkan-industri-akuakultur/full&view=ok>. [Accessed Sep. 23, 2020].
- [3] F. Gheost, "Parameter Kualitas Air berdasarkan Faktor Kimia, Fisika dan Biologinya," *Geologinesia*, 2018. [Online]. Available: <https://www.geologinesia.com/2018/08/parameter-kualitas-air.html>. [Accessed Sep. 23, 2020].
- [4] A. Junaidi and C. Kartiko, "Design of Pond Water Quality Monitoring System Based on Internet of Things and Pond Fish Market in Real-Time to Support the Industrial Revolution 4.0," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 771, no. 1, pp. 0–6, 2020, doi: 10.1088/1757-899X/771/1/012034.
- [5] Z. Zainuddin, R. Idris, and A. Azis, "Water Quality Monitoring System for Vannamae Shrimp Cultivation Based on Wireless Sensor Network In Taipa, Mappakasunggu District, Takalar," *ICMEMe 2018*, vol. 165, no. January, pp. 89–92, 2019, doi: 10.2991/icmeme-18.2019.20.
- [6] Q. Lin *et al.*, "Deep-learning based approach for forecast of water quality in intensive shrimp ponds," *Indian J. Fish.*, vol. 65, no. 4, pp. 75–80, 2018, doi: 10.21077/ijf.2018.65.4.72559-09.
- [7] F. Umam and H. Budiarto, "Water Quality Control for Shrimp Pond Using Adaptive Neuro Fuzzy Inference System: The First Project," *J. Phys. Conf. Ser.*, vol. 953, no. 1, pp. 1–5, 2018, doi: 10.1088/1742-6596/953/1/012134.
- [8] A. Sahrijanna and Sahabuddin, "Kajian Kualitas Air Pada Budi Daya Udang Vaname Dengan Sistem Pergiliran Pakan di Tambak Intensif," *Pros. Forum Inov. Teknol. Akuakultur 2014*, pp. 329–336, 2014.