

Monitoring Kualitas Air pada Budidaya Udang

Anggota Kelompok:

- Abdul Ghofar Alhasyim
- Larasati
- Richard Alvin Pratama
- Tezard Almafazan Mulia

Background

Problem, Validation, and Literature



Problem

Banyak pembudidaya udang jenis vaname, sering mengalami gagal panen namun tidak mengetahui secara pasti apa yang menjadi penyebabnya.



Solution

Membuat sistem IoT yang selalu memonitor kualitas air pada kolam budidaya

Pembudidaya dapat dengan mudah mengetahui kondisi kualitas air serta langkah penanganan ketika terjadi perubahan.

Problem & Solution



Pemerintah Kabupaten

Bapak Suparjo,
S.Tr.Pi. selaku
penyuluhan perikanan
Dinas Perikanan
Kabupaten
Pemalang



Pengaruh Kualitas Air pada Udang



PH dan Suhu sebagai Indikator

Solid Literature

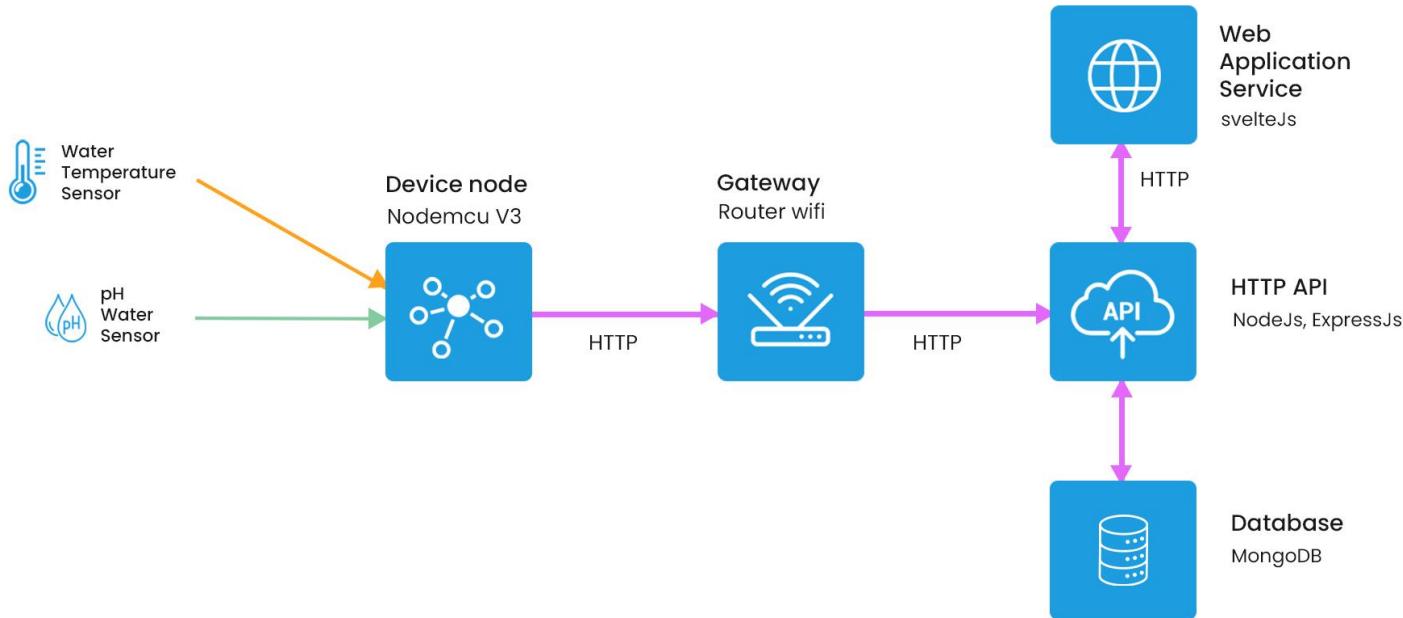
References Document :

https://drive.google.com/drive/folders/1LBsk_UfUC3LNpFqXHfq-lohnkwikPr3y?usp=sharing

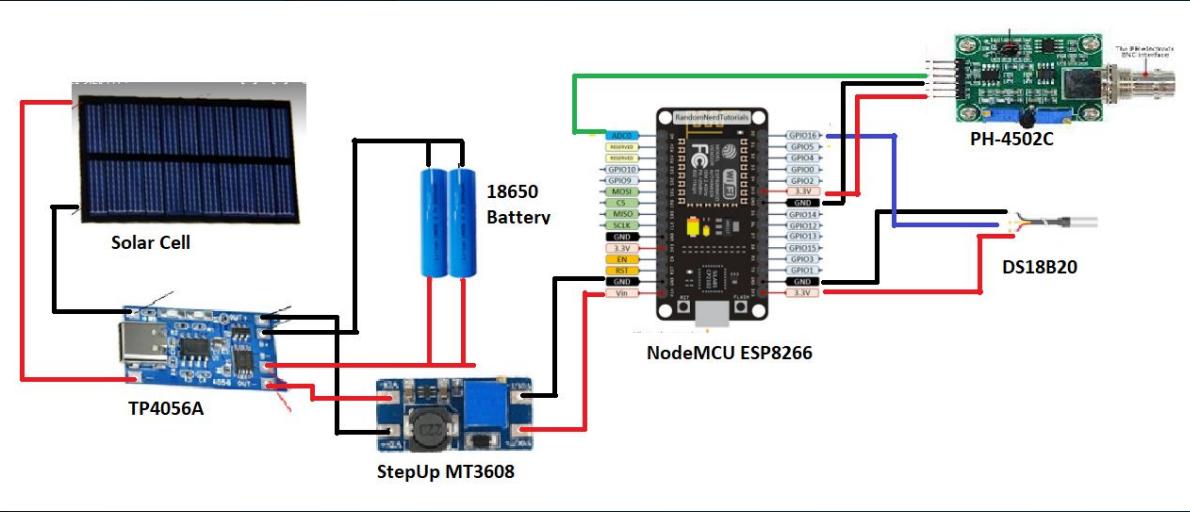
System Architecture & Demo

Backend, Database, Frontend,
Authentication, and Authorization

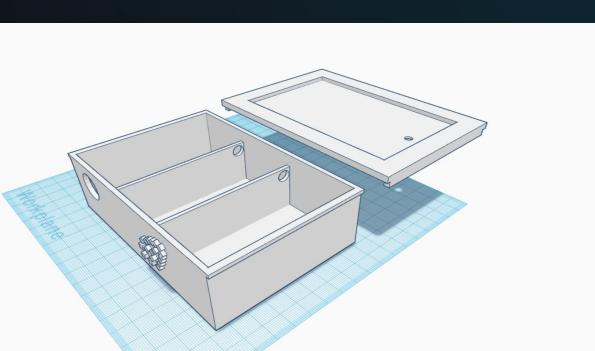
General Architecture



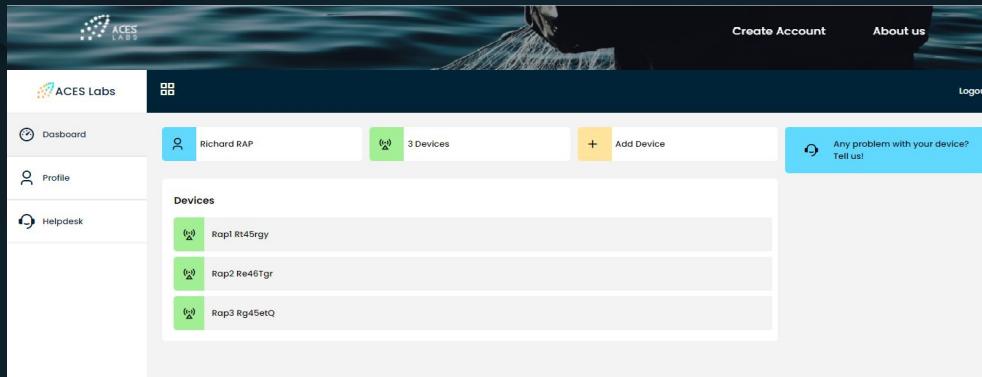
Devices Architecture & Design



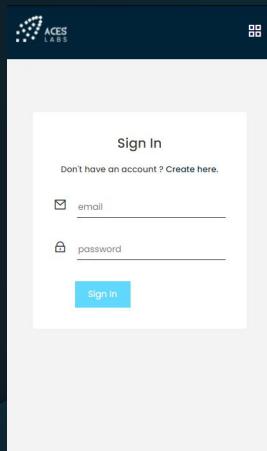
- Nodemcu V3 ESP8266
- DS18B20
- PH-4502C
- TP4056A
- Step Up MT3608
- Solar Cell
- I8650 Battery



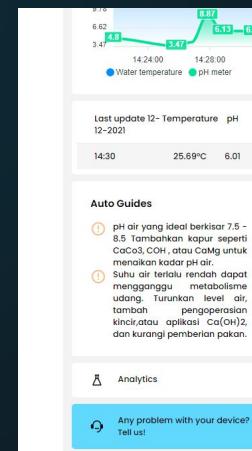
Frontend



The dashboard shows a user profile for "Richard RAP" with 3 devices. The devices listed are Rap1 Rt45rgy, Rap2 Re46tgr, and Rap3 Rg45etQ. A blue button "Add Device" is available. A feedback section asks if there's any problem with the device.



Sign In
Don't have an account? Create here.
Email: _____
Password: _____
Sign in



APEXCHARST

SVELTE

Backend

Dotenv

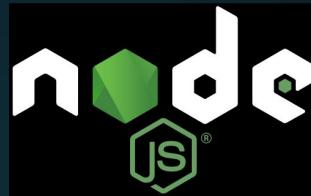
Bcryptjs

Joi

Cors

jsonwebtoken

socket.io



Express **JS**

utils > **JS** db.js > ...

```
1 const mongoose = require('mongoose');
2 // const dotenv = require('dotenv');
3 // dotenv.config();
4 mongoose.connect(process.env.DB_CONNECT,
5
```

.env

```
1 TOKEN_SECRET=[REDACTED]
2 DB_CONNECT=mongodb+srv://admin-
3
```

controller
JS commentController.js
JS histController.js
JS userController.js
model
JS comment.js
JS hist_kolam.js
JS user.js
> node_modules
> routes

List API backend

No	API	Method	Input
1	https://ghorilard.herokuapp.com/	GET	Null
2	https://ghorilard.herokuapp.com/register/	GET	Null
3	https://ghorilard.herokuapp.com/register/	POST	nama_user, email_user, password
4	https://ghorilard.herokuapp.com/login/	GET	Null
5	https://ghorilard.herokuapp.com/logout/	POST	Null

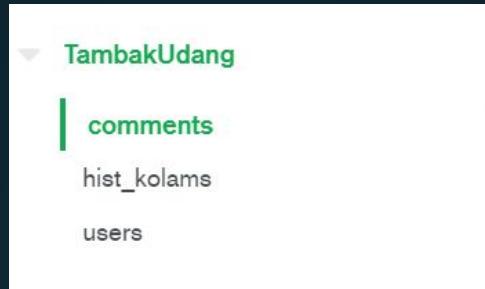
<http://ghorilard.herokuapp.com/>

Database



Users

```
_id: ObjectId("615aa6cfbe948ed6f4255662")
nama_user: "Richard RAP"
email_user: "richard.alvin@student.umn.ac.id"
password: "$2a$10$IW53E8dtIZ2hTbW/qLYMeryLWYq0UpereIhfrUcnfzNKZedMHv6"
auth: 2
data_kolam: Array
  0: Object
    nama_kolam: "Rap1"
    kode_device: "Rt45rgy"
    link_history: "https://ghorilard.herokuapp.com/hist/Rt45rgy"
    timestamp: 2021-10-04T07:01:35.513+00:00
    _id: ObjectId("615aa6cfbe948ed6f4255663")
  > 1: Object
  > 2: Object
tg1_buatakun: 2021-10-04T07:01:35.513+00:00
  ...
```



hist_kolams

```
_id: ObjectId("61a3157b9372276387ef9f7b")
id_kolam: "Re46Tgr"
relay: false
token: "$2a$10$TMQ/dhKhZoNI.31TZKYI10qk5buPMwd5/3j
data_hist: Array
  0: Object
    water_temp: 25.19
    ph_meter: 4.8
    timestamp: 2021-12-12T07:21:46.956+00:00
    _id: ObjectId("61b5a30ad69a19ff108fbff8")
  ...
```

Comments

```
_id: ObjectId("6185604cf3d73159270d96d1")
data_comment: Array
  0: Object
    desc_comment: "Rasengan chidori"
    tgl_comment: 2021-11-05T16:48:12.951+00:00
    _id: ObjectId("6185604cf3d73159270d96d2")
  > 1: Object
    __v: 1
    id_user: "618559878a70c447c83ab979"
```

Authentication & Authorization

- Password user di hash menggunakan bcryptjs + random salt
- User harus login untuk bisa akses API yang terproteksi
- User dan Admin punya privilege berbeda
- User tidak bisa akses user lain

```
//js verifyToken.js > <unknown> > exports
const jwt = require('jsonwebtoken');

module.exports = (req,res,next)=>{
  const token = req.header('token');
  if(!token) return res.status(401).send('Access Denied');
  try{
    const verified = jwt.verify(token,process.env.TOKEN_SECRET);
    req.user = verified;
    if(req.user._id === req.headers._id || req.user.auth === 1){
      next();
    }else if(req.user._id === req.headers._id){
      next();
    }else{
      res.status(401).send('Access Denied');
    }
  }catch(err){
    res.status(400).send('Invalid Token');
  }
}
```

```
//password is correct
const validPass = await bcrypt.compare(req.body.password, user.password);
//create and assign a token
if(user.auth === 1){
  token = jwt.sign({_id: user._id, auth: 1, exp: Math.floor(Date.now()/1000)+(60*60)},
```

Authentication & Authorization

- Devices terdapat token
- Token setiap devices terdapat pada setiap database
- Backend akan membandingkan token device dan token pada database

```
26  
27 //HTTP  
28 String token = "--"; //Token device  
29 String serverName = "--"; //URL API  
30
```

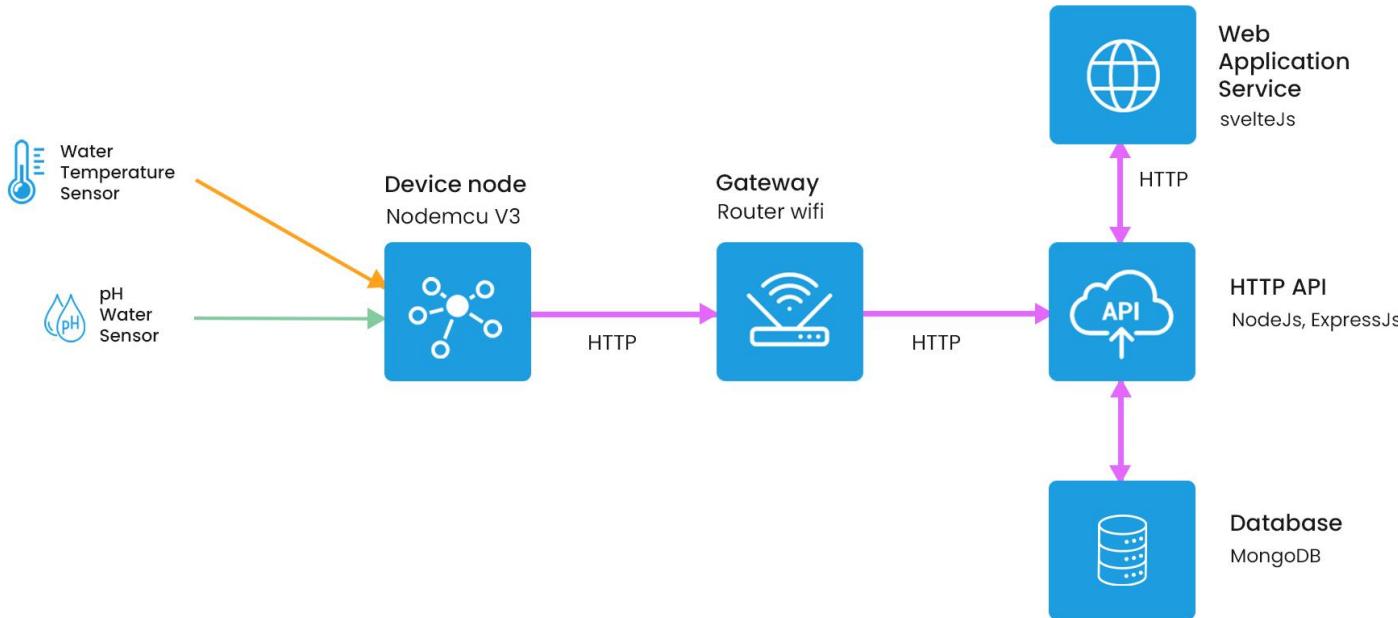
```
_id: ObjectId("61a3157b9372276387ef9f7b")  
id_kolam: "Re46Tgr"  
relay: false  
token: "$2a$10$TMQ/dhKhZoNI.31TZKYIl0qk5buPMwd5/3yieG8JwteyiVUKobtGe"  
> data_hist: Array
```

```
if(Hist_kolam1 == null){  
    const salt = await bcrypt.genSalt(10);  
    const hashToken = await bcrypt.hash(req.body.token, salt);  
  
}  
else{  
    const validPass = await bcrypt.compare(req.body.token, Hist_kolam1.token);  
    if(validPass){  
        res.json({  
            status: true,  
            message: "Success",  
            data: Hist_kolam1  
        })  
    }  
}
```

Communication Protocol

How elements communicate,
communication flow, and message
format

Elements Communicate, Flow, & Format



Cloud Deployment

Cloud services & how they deploy

Cloud Services



Frontend

- Gratis
- Interface Intuitive, ease to use.
- Setting env variable straight forward



Backend

- Gratis
- Banyak dokumentasi
- Bisa masukkan config vars



Database

- Gratis
- Bawaan mongodb

How To Deploy



Push project frontend ke repository github

Login akun Vercel www.vercel.com

Hubungkan repository frontend dengan project vercel

Build app

How To Deploy



heroku

- Download Heroku CLI
- heroku login dan pastikan node, npm, dan git sesuai
- heroku create
- git push heroku main



Atlas

- Buat cluster dan buat database
- membuat user dengan privilege
- setting network address
- connect to your application

The screenshot shows the Heroku dashboard interface. At the top, there's a purple header bar with the text "5 Profile" and "1 web: node app.js". Below the header, the user profile "ghorilard" is shown with a purple icon. The dashboard has tabs for "Overview", "Resources", "Deploy", and "Metrics". The "Deploy" tab is currently active.

This screenshot shows the "Config Vars" section of the Heroku deployment configuration. It lists three environment variables: "DB_CONNECT", "TOKEN_SECRET", and "KEY". Each variable has a "Value" field next to it, which is currently redacted with a black rectangle.

This screenshot shows the "Select your driver and version" step of the MongoDB connection setup. It shows a dropdown menu where "Node.js" is selected under the "DRIVER" column, and "4.0 or later" is selected under the "VERSION" column. Below this, the "Add your connection string into your application code" step is shown, containing a text input field with the connection string: "mongodb+srv://admin-alvin:<password>@cluster0.sygau.mongodb.net/myFirstDatabase?retryWrites=true&w=majority". There is also a checkbox for "Include full driver code example".

DEMO

Thank You

Ghorilard

Connect All For Convenience