Monitoring Kualitas Air pada Budidaya Udang

Anggota Kelompok:

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

Background

Problem, Validation, and Literature

Problem & Solution

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.

Pemerintah Kabupaten

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

Pengaruh Kualitas Air pada Udang

PH dan Suhu sebagai Indikator

Solid Literature

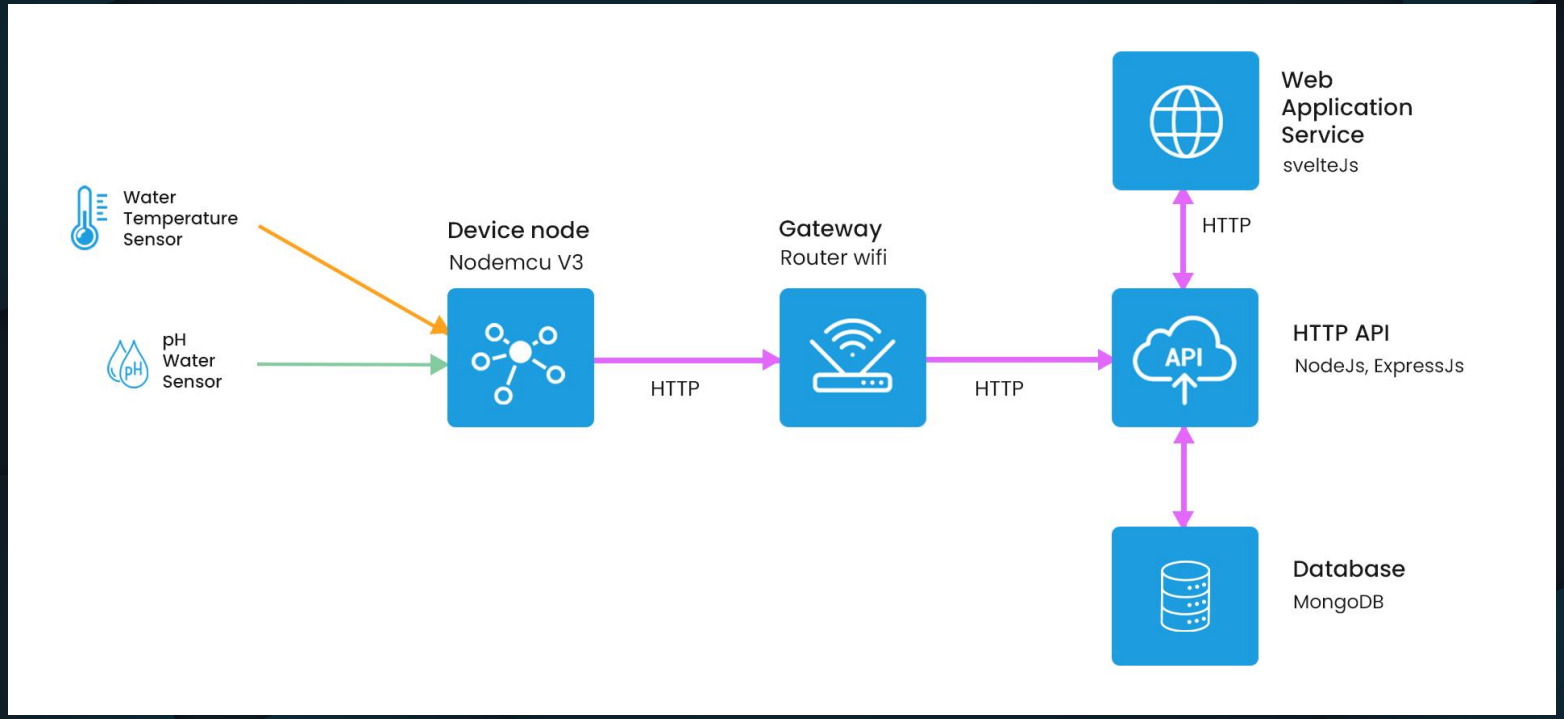
References Document :

https://drive.google.com/drive/folders/1LBsk_UfUC3LNpFqXHfq-IohnkwikPr3y?usp=sharing

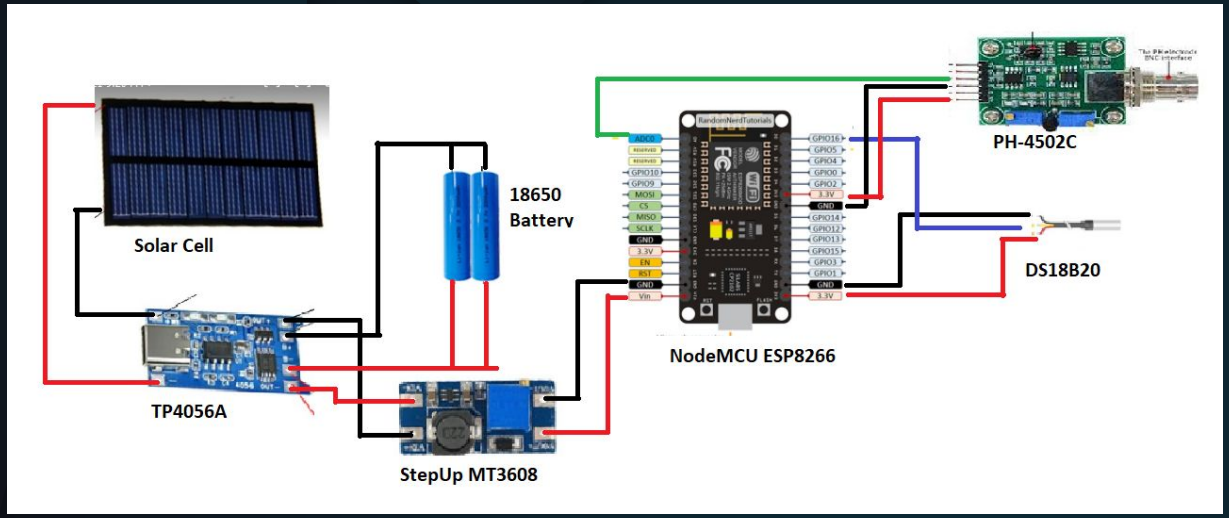
System Architecture & Demo

Backend, Database, Frontend,
Authentication, and Authorization

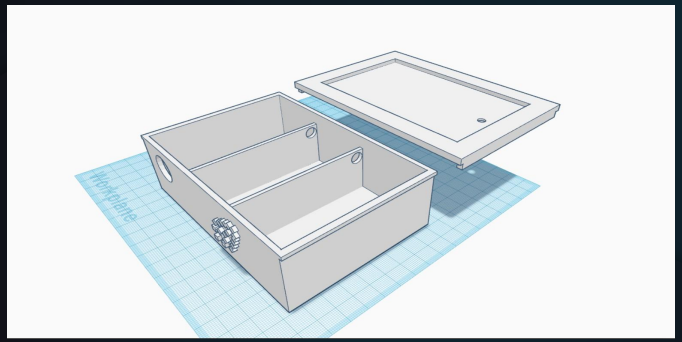
General Architecture

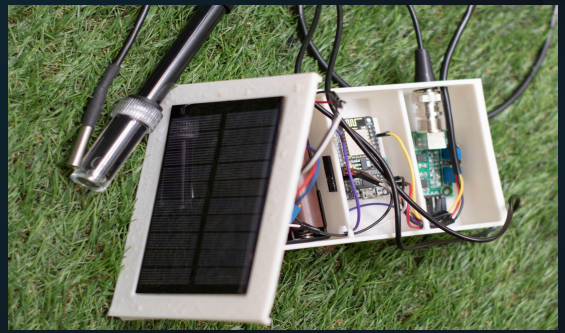


Devices Architecture & Design

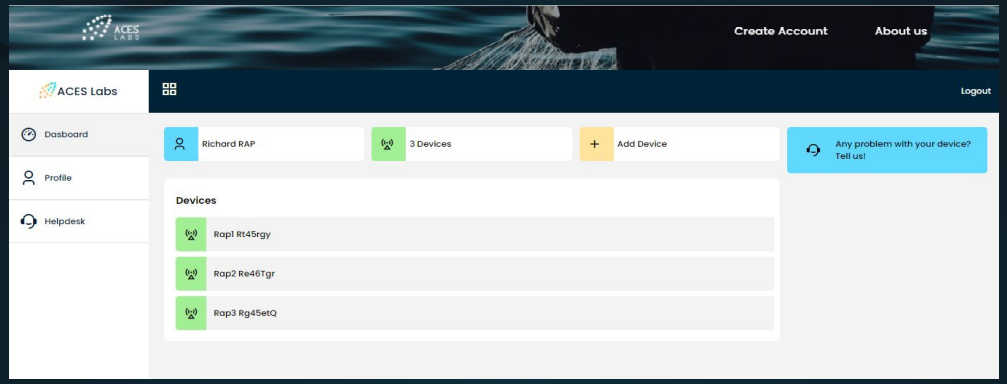


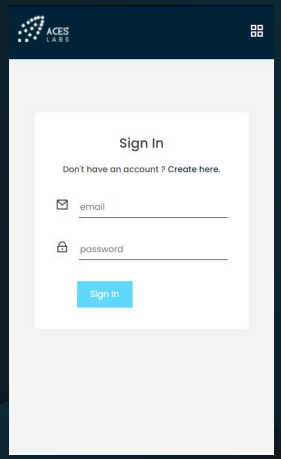
- Nodemcu V3 ESP8266
- DS18B20
- PH-4502C
- TP4056A
- Step Up MT3608
- Solar Cell
- 18650 Battery

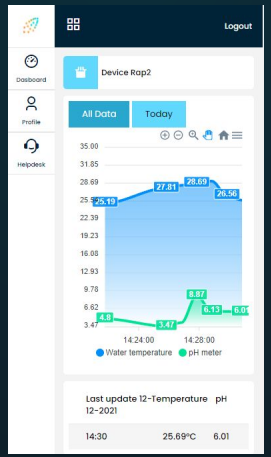


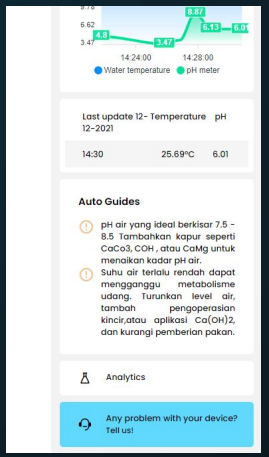


Frontend









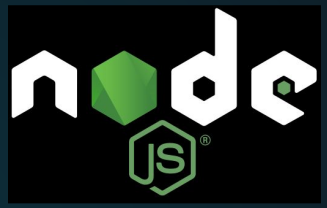
APEXCHARST

SVELTE

Backend

Dotenv | Bcryptjs | Joi

Cors | jsonwebtoken | socket.io

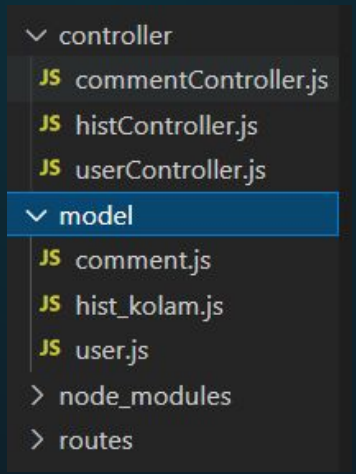


utils > JS db.js > ...

```
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
4
```

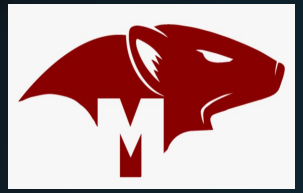


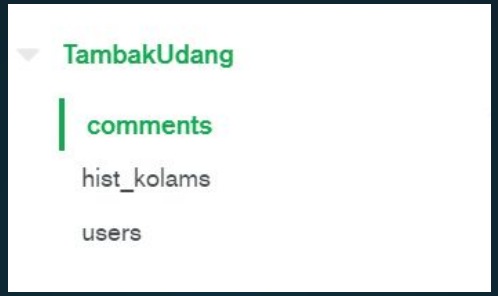
List API backend

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, passwo
4	https://ghorilard.herokuapp.com/login/	GET	Null

http://ghorilard.herokuapp.com/

Database

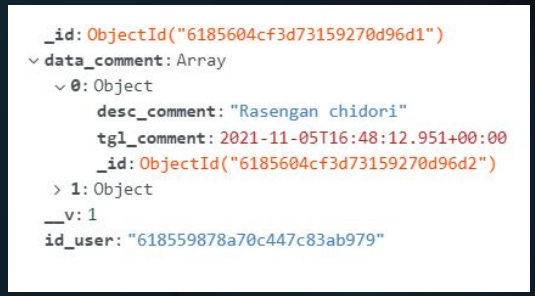




Users

hist_kolams

Comments



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

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

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

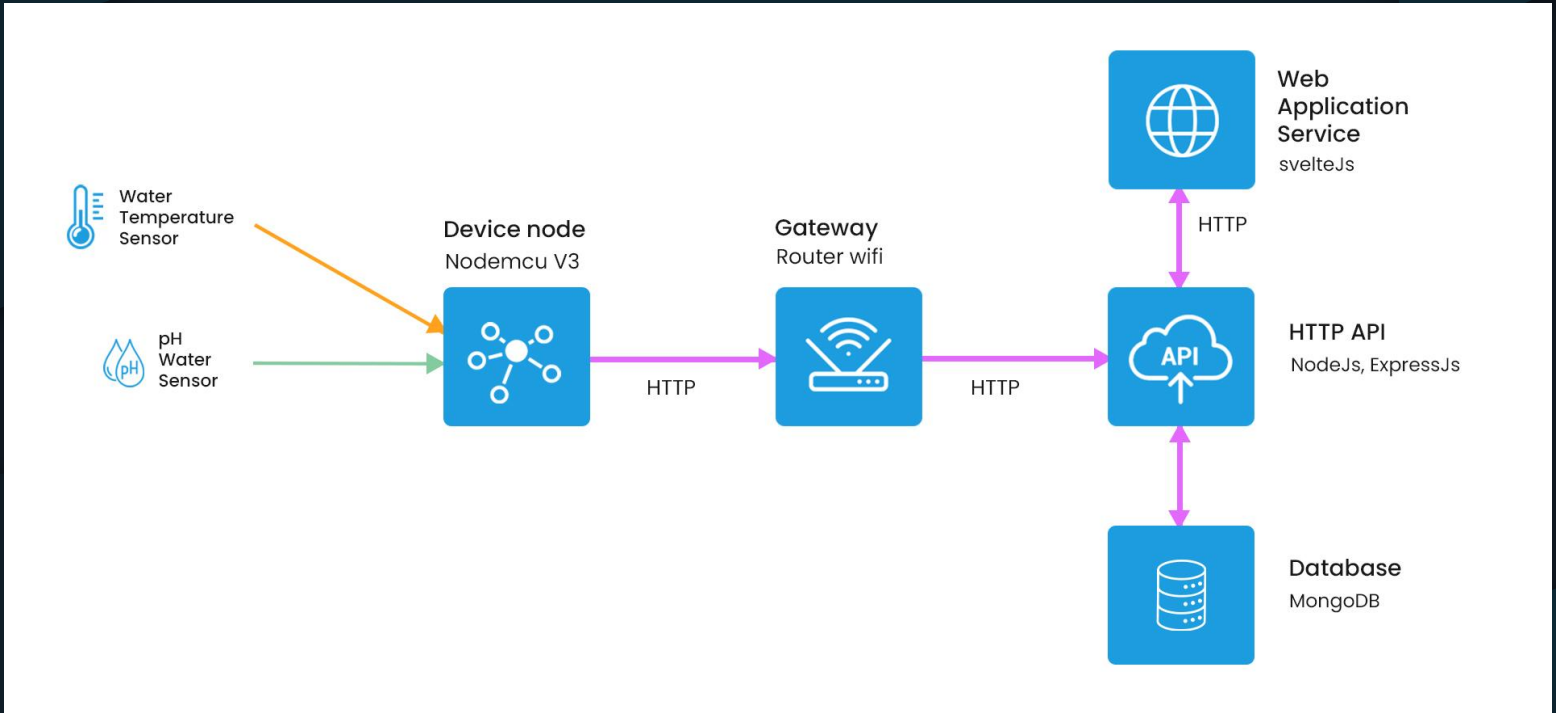

```
if(Hist_kolam1 == null){
    const salt = await bcrypt.genSalt(10);
    const hashToken = await bcrypt.hash(req.body.token, salt);
```

```
}else{
    const validPass = await bcrypt.compare(req.body.token, Hist_kolam1.token);
    if(validPass){
```

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

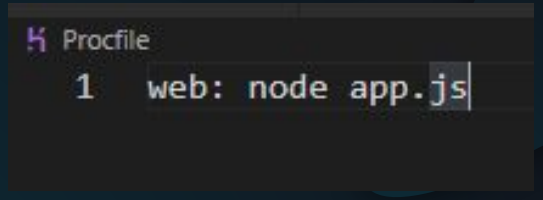
Login akun Vercel www.vercel.com

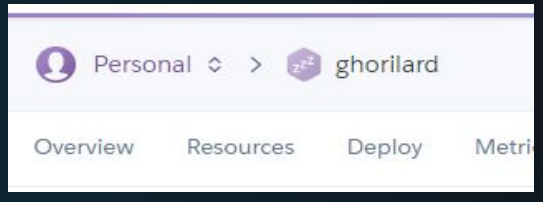
Hubungkan repository frontend dengan project vercel

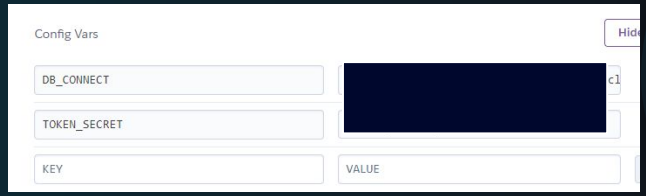
Build app

How To Deploy

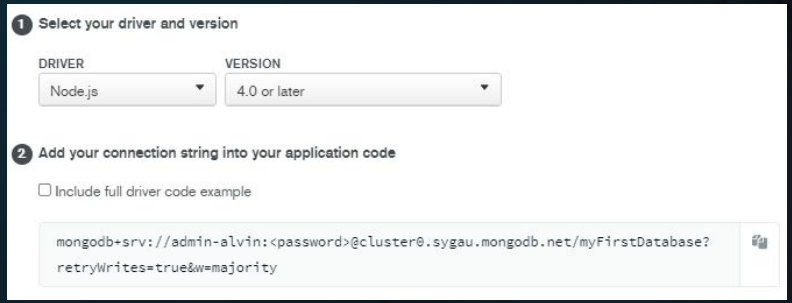
- Download Heroku CLI
- heroku login dan pastikan node, npm, dan git sesuai
- heroku create
- git push heroku main







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



DEMO

Thank You

Ghorilard

Connect All For Convenience