

**PROTOTYPE KONTROL DAN MONITORING SYSTEM PENGISIAN BAHAN BAKAR  
PADA GENSET MENGGUNAKAN RASPBERRY PI DAN TAMPILAN LCD DI  
BANDARA**

**TUGAS AKHIR**



Oleh :

**MUHAMAD ARYAFATA KUSUMA**  
**NIT. 30118016**

**PROGRAM STUDI DIPLOMA 3 TEKNIK LISTRIK BANDARA POLITEKNIK  
PENERBANGAN SURABAYA  
2021**

**PROTOTYPE KONTROL DAN MONITORING SYSTEM PENGISIAN BAHAN BAKAR  
PADA GENSET MENGGUNAKAN RASPBERRY PI DAN TAMPILAN LCD DI  
BANDARA**

**TUGAS AKHIR**

Diajukan sebagai Salah Satu Syarat untuk Mendapatkan Gelar Ahli Madya  
(A.Md.). Pada Program Studi Diploma III Teknik Listrik Bandar Udara



Oleh :

**MUHAMAD ARYAFATA KUSUMA**

**NIT. 30118016**

**PROGRAM STUDI DIPLOMA III TEKNIK LISTRIK BANDAR UDARA  
POLITEKNIK PENERBANGAN SURABAYA**

**2021**

## LEMBAR PERSETUJUAN


PROTOTYPE KONTROL DAN MONITORING SYSTEM PENGISIAN  
BAHAN BAKAR PADA GENSET MENGGUNAKAN RASPBERRY PI DAN  
TAMPILAN LCD DI BANDARA

Oleh :

Muhamad Aryafata Kusuma  
NIT. 30118016

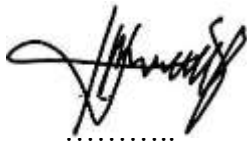
Disetujui untuk diujikan pada :  
Surabaya, Agustus 2021

Pembimbing I : SLAMET HARIYADI, ST, MM  
NIP. 19630408 198902 1 001



.....

Pembimbing II : Dr. WIWID SURYONO, S.Pd, MM  
NIP. 19611130 198603 1 001



.....

## LEMBAR PENGESAHAN

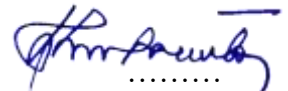
### PROTOTYPE KONTROL DAN MONITORING SYSTEM PENGISIAN BAHAN BAKAR PADA GENSET MENGGUNAKAN RASPBERRY PI DAN TAMPILAN LCD DI BANDARA

Oleh :  
Muhamad Aryafata Kusuma  
NIT. 30118016

Telah dipertahankan dan dinyatakan lulus pada Ujian Tugas Akhir Program  
Pendidikan Diploma 3 Teknik Listrik Bandara  
Politeknik Penerbangan Surabaya  
Pada tanggal : Agustus 2021

#### Panitia Penguji :

1. Ketua : Dr. SUHANTO, S.Kom.,MM  
NIP. 198005082002121003
2. Sekretaris : HERY ISMIANTO,ST,MM
3. Anggota : SLAMET HARIYADI, ST,.MM  
NIP. 19630408 198902 1 001



.....



.....



.....

Ketua Program Studi  
D3 Teknik Listrik Bandara

RIFDIAN IS., ST, MT.  
NIP. 19810629 200912 1 002

## **ABSTRAK**

### **PROTOTYPE KONTROL DAN MONITORING SYSTEM PENGISIAN BAHAN BAKAR PADA GENSET MENGGUNAKAN RASPBERRY PI DAN TAMPILAN LCD DI BANDARA**

OLEH :

M ARYAFATA KUSUMA

Masih banyaknya penggunaan mesin pompa air secara manual dan sering mengakibatkan meluapnya bahan bakar solar ketika tangki sudah terisi penuh akibat kurangnya pengontrolan pada saat pengisian maka penulis membuat sistem kontrol dan monitoring system pengisian bahan bakar pada genset menggunakan raspberry pi. Penelitian ini bertujuan untuk memonitoring ketinggian air dan juga sistem kontrol pompa secara otomatis dengan menggunakan sensor ultrasonik yang akan mendeteksi ketinggian air pada tangki. Penelitian ini menggunakan Raspberry pi karena dapat memudahkan dalam melakukan pengontrolan dari jarak jauh tanpa harus datang ke lokasi terlebih dahulu. Dalam penelitian ini juga menggunakan link yang akan dihubungkan dengan web yang ada di android atau lain sebagainya agar perancangan ini dapat digunakan sesuai keinginan. Alat ini juga menggunakan sensor ultrasonik yang akan mendeteksi ketinggian air pada tangki sehingga pada saat pengisian air tidak sampai melebihi batas maksimal. Batas maksimal untuk alat ini yaitu 80%, dikarenakan apabila terisi 100% maka akan mengakibatkan sensor ultrasonik terendam dan sensor tersebut akan mengalami rusak serta tidak berfungsi dengan normal. Hasil dari penelitian ini adalah kita dapat mengetahui kondisi dan juga jumlah ketinggian air dalam bentuk persentase dari jarak jauh tanpa harus mengecek langsung dan cukup dengan mengontrol lewat handphone atau lain sebagainya yang bisa terhubung ke internet.

**Kata kunci** : Generator set (genset), Raspberry, *Wi-Fi*, Pompa, Sensor volumebahan bakar, Bandar Udara.

## **ABSTRACT**

### **PROTOTYPE KONTROL DAN MONITORING SYSTEM PENGISIAN BAHAN BAKAR PADA GENSET MENGGUNAKAN RASPBERRY PI DAN TAMPILAN LCD DI BANDARA**

OLEH :

M ARYAFATA KUSUMA

There is still a lot of use of water pumping machines manually and often results in the overflow of diesel fuel when the tank is full due to lack of control at the time of filling, the authors make a control system and monitoring system for refueling the generator using a raspberry pi. This study aims to monitor the water level and also the pump control system automatically using ultrasonic sensors that will detect the water level in the tank. This research uses Raspberry pi because it can make it easier to control remotely without having to come to the location first. In this study also uses a link that will be connected to the web on Android or so on so that this design can be used as desired. This tool also uses an ultrasonic sensor that will detect the water level in the tank so that when filling the water it does not exceed the maximum limit. The maximum limit for this tool is 80%, because if it is filled 100% it will result in the ultrasonic sensor being submerged and the sensor will be damaged and will not function normally. The results of this study are that we can find out the condition and the amount of water level in the form of a percentage from a distance without having to check directly and simply by controlling it via cellphone or other things that can be connected to the internet.

**Keywords:** Generator set (genset), Raspberry, Wi-Fi, Pump, Fuel volume sensor, Airport.

## PERNYATAAN KEASLIAN DAN HAK CIPTA

Saya yang bertanda tangan di bawah ini :

Nama : Muhamad Aryafata kusuma  
NIT : 30118016  
Program Studi : D-III Teknik Listrik Bandar Udara  
Judul Tugas Akhir : Prototype kontrol dan monitoring system pengisian bahan bakar pada genset menggunakan raspberry pi dan tampilan LCD di bandara.

dengan ini menyatakan bahwa :

1. Tugas Akhir ini merupakan karya asli dan belum pernah diajukan untuk mendapatkan gelar akademik, baik di Politeknik Penerbangan Surabaya maupun di Perguruan Tinggi lain, serta dipublikasikan, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
2. Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan Hak Bebas Royalti Non Eksklusif (*Non-Exclusive Royalty-Free Right*) kepada Politeknik Penerbangan Surabaya beserta perangkat yang ada (jika diperlukan). Dengan hak ini, Politeknik Penerbangan Surabaya berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya dengan tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya. Apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Politeknik Penerbangan Surabaya.

Surabaya, 03 Agustus 2021



Muhamad Aryafata kusuma  
NIT.30118026

**MOTTO**

**“ NEVER STOP TRYING ”**



## KATA PENGANTAR

Puji syukur kehadirat ALLAH SWT atas segala rahmat dan karunia – Nya yang telah memberikan kesehatan, pengetahuan, keterampilan, pengalaman yang senantiasa diberikan kepada penulis, sehingga penulis bisa menyelesaikan Laporan Tugas Akhir yang berjudul **“PROTOTYPE KONTROL DAN MONITORING SYSTEM PENGISIAN BAHAN BAKAR PADA GENSET MENGGUNAKAN RASPBERRY PI DAN TAMPILAN LCD DI BANDARA”** dengan baik dan lancar sesuai dengan waktu yang ditetapkan dan sebagai syarat untuk menyelesaikan program Diploma 3 Teknik Listrik Bandara di Politeknik Penerbangan Surabaya.

Selama proses penyusunan tugas khusus ini penulis banyak menerima bantuan, bimbingan dan pengarahan dari berbagai pihak baik material spiritual, materi serta saran. Pada kesempatan ini mengucapkan terima kasih kepada:

1. Orang tua yang selalu mendukung saya dan memberi semangat, menjadi motivasi yang membuat semangat tanpa batas dalam berusaha dan bekerja.
2. Bapak M Andra Adityawarman, ST, MT. selaku Direktur Politeknik Penerbangan Surabaya.
3. Bapak Rifdian I.S.,ST, MT. selaku Ketua Program Studi Teknik Listrik Bandara Politeknik Penerbangan Surabaya.
4. Bapak Slamet Hariyadi, ST, MM. selaku Dosen Pembimbing Materi Tugas Akhir.
5. Bapak Dr. Wiwid Suryono, S.Pd, MM selaku Pembimbing Materi penulisan Tugas Akhir.
6. Dosen Politeknik Penerbangan Surabaya yang telah memberikan pengetahuan dan memberikan pelajaran yang berharga untuk penulis serta teman–teman Teknik Listrik Bandara angkatan XIII yang telah memberikan banyak bantuan, support dan motivasi.

Penulis menyadari bahwa penulisan ini masih jauh dari sempurna dan masih banyak kekurangan, maka dari itu penulis mengharapkan saran dan kritik yang dapat membangun dari semua pihak agar dapat membantu untuk menjadikan penulisan Tugas Akhir selanjutnya lebih baik.

Akhir kata penulis berharap semoga penulisan ini dapat bermanfaat dan selanjutnya dapat dikembangkan, berguna bagi semua pihak.

Surabaya, Agustus 2021

Penulis

## DAFTAR ISI

TUGAS AKHIR .....	2
LEMBAR PERSETUJUAN .....	iii
LEMBAR PENGESAHAN .....	iv
ABSTRAK .....	iv
ABSTRACT.....	vi
MOTTO .....	viii
KATA PENGANTAR.....	ix
BAB 1 .....	1
PENDAHULUAN.....	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan masalah .....	2
1.4 Tujuan Penelitian .....	3
1.5 Manfaat Penelitian .....	3
1.6 Sistematika Penulisan .....	3
BAB 2 .....	5
2.1 Teori Penunjang .....	5
2.1.1 Relay .....	5
2.1.2 Motor induksi satu phase .....	6
2.1.3 Pompa .....	9
2.1.4 Generator Set .....	10
2.1.5 Tanki Bahan Bakar .....	11
2.1.6 Raspberry pi.....	12
2.1.7 Sensor.....	12
2.2 Kajian Penelitian Terdahulu Yang Relevan.....	14
BAB 3 .....	41
3.1 Desain Penelitian .....	41
3.2 Perancangan Alat .....	41
3.2.1 Desain Alat .....	41
3.2.2 Cara Kerja Alat .....	43

3.2.3	Komponen Alat .....	44
3.3	Teknik Pengujian.....	46
3.4	Teknik Analisis Data .....	46
3.5	Tempat dan Waktu Penelitian.....	47
BAB IV	.....	48
4.1	Hasil Penelitian Perangkat Keras ( <i>Hardware</i> ) .....	48
4.1.1	Pengujian dan Analisa Adaptor <i>Power Supply</i> .....	48
4.1.2	Pengujian dan analisa Raspbery pi.....	49
4.1.3	Pengujian dan Analisis Pompa Air .....	52
4.1.4	Pengujian dan Analisa Modul <i>Relay</i> .....	53
4.1.5	Pengujian dan Analisa Sensor Ultrasonik HC-SR04.....	54
4.1.6	Pengujian dan Analisa Sensor <i>Flow Meter</i> .....	56
4.1.7	Hasil Analisa Koneksi Wifi .....	58
4.2	Hasil Penelitian Perangkat Lunak .....	59
4.2.1	Pengujian dan Analisa Aplikasi Kontrol dan Monitoring pada System pengisian bahan bakar otomatis.....	59
4.3	Sistem Alat Keseluruhan.....	61
4.2.2	Validasi... ..	64
4.4	Pembahasan Hasil Penelitian.....	69
4.2.3	Kelebihan Alat.....	69
4.2.4	Kekurangan Alat.....	69
BAB V	.....	70
PENUTUP	.....	70
5.1	Kesimpulan .....	70
5.2	Saran.....	70
DAFTAR PUSTAKA	.....	72



## DAFTAR GAMBAR

Gambar 2. 1 Prinsip Medan Magnet Utama dan Medan Magnet Bantu .....	7
Gambar 2. 2 Grafik gelombang arus medan bantu dan arus medan utama.....	7
Gambar 2. 3 Medan magnet pada Stator Motor satu fasa .....	8
Gambar 2. 4 Rotor sangkar .....	8
Gambar 2. 5 Pompa dinamis .....	10
Gambar 2. 6 Tanki Genset.....	12
Gambar 2. 7 Raspberry pi .....	12
Gambar 2. 8 Flow sensor.....	14
Gambar 2. 9 Ultrasonic sensor ( HC-SR04 ).....	14
Gambar 3. 1 Diagram blok.....	16
Gambar 3. 2 Flowchart .....	18
Gambar 3. 3 PythonRaspberry Software .....	20
Gambar 3. 4 Tampilan kontrol dan monitoring.....	21

## DAFTAR PUSTAKA

- Bunga, P., Pakiding, M., & Silimang, S. (2015). Perancangan sistem pengendalian beban dari jarak jauh menggunakan smart relay. *Jurnal Teknik Elektro Dan Komputer*, 4(5), 65–75.
- Hasto, K., Haddin, M., & Nugroho, D. (2015). Kendali arus starting motor induksi satu fasa menggunakan magnetic energy recovery switch (Mers). *Media ElektriKa*, 8(2).
- MUHAMMAD ALIF, S. (2019). PENTINGNYA GENERATOR UNTUK MEMENUHI KEBUTUHAN ENERGI LISTRIK DI KAPAL MOTOR SINAR BANGUN. *KARYA TULIS*.
- PANGESTIKA, M. A. Y. U. (2017). *IMPLEMENTASI RASPBERRY PI SEBAGAI PENGUBAH SUARA MENJADI TEKS*. POLITEKNIK NEGERI SRIWIJAYA.
- Ramadhan, D., Serevina, V., & Raihanati, R. (2016). Pengembangan alat praktikum viskometer metode bola jatuh bebas berbasis sensor efek hall UGN3503 sebagai media pembelajaran fisika. *Prosiding Seminar Nasional Fisika (E-Journal)*, 5, SNF2016-RND.
- Sofyan, M. I. (2018). KONSUMSI BAHAN BAKAR SPESIFIK GENSET KUBOTA J310 BERBAHAN BAKAR CRUDE-OIL DENGAN PEMBANDING BIO-SOLAR PADA KONDISI BEBAN 100%. *Jurnal Energi Dan Lingkungan (Enerlink)*, 14(1).
- Yudha, P. S. F., & Sani, R. A. (2017). Implementasi Sensor Ultrasonik Hc-Sr04 Sebagai Sensor Parkir Mobil Berbasis Arduino. *EINSTEIN (e-Journal)*, 5(3).

## **LAMPIRAN 1**

### **PROTOTYPE KONTROL DAN MONITORING SYSTEM PENGISIAN BAHAN BAKAR PADA GENSET MENGGUNAKAN RASPBERRY PI DAN TAMPILAN LCD DI BANDARA**

Oleh :  
M Aryafata Kusuma  
(30118016)

Standar Operasional Prosedur (SOP) merupakan pedoman atau panduan bagaimana melakukan suatu pekerjaan atau menggambarkan serangkaian instruksi yang harus di lakukan.

Berikut ini merupakan prosedur untuk menggunakan alat “PROTOTYPE KONTROL DAN MONITORING SYSTEM PENGISIAN BAHAN BAKAR PADA GENSET MENGGUNAKAN RASPBERRY PI DAN TAMPILAN LCD DI BANDARA”

1. SOP ( Standart Operasional Prosedure ) Penggunaan prototype pengisian bahan bakar otomatis.
  - Sambungkan wifi ke telepon genggam sesuai dengan ssid dan juga password yang telah di atur di dalam Raspberry pi.
  - Pastikan koneksi internet dalam kondisi stabil agar tidak terjadi delay saat menjalankan prototype pengisian bahan bakar otomatis tersebut.



- Setelah internet benar benar terkoneksi dengan benar, buka aplikasi pada telepon genggam untuk mengaktifkan pompa pintar kemudian Log in dengan user name dan password yang sudah di atur sebelumnya.



- Sambungkan daya untuk alat
- Sambungkan daya untuks raspberry dan sensor ultrasonik

- Pilih mode auto dan juga manual



- Jika pada mode auto maka pompa tersebut akan otomatis menyala jika sensor ultrasonik mendeteksi level ketinggian air telah mencapai di bawah 20% dan pompa akan mati sendiri jika sensor ultrasonik telah mencapai atau diatas 20% .
- Jika pada mode manual maka kontrol untuk pompa sendiri dilakukan secara manual melalui aplikasi pompa pintar tersebut, beserta tampilan monitoring yang akan di tampilkan pada aplikasi tersebut berupa level ketinggian air dan besarnya aliran air yang melalui sensor Flowmeter yang di pasang pada pipa penyambung antara box plastik satu dengan yang lain sebagai simulasi dari Main

Tank dan Tangki Genset.

## LAMPIRAN 2

Data sheet water flow sensor



Working voltage	5V-24V
Maximum current	15 mA (DC 5V)
Weight	43 g
External diameters	20mm
Flow rate range	1 ~ 30 L/min
Operating temperature	0°C~80°C
Liquid temperature	<120°C
Operating humidity	35% ~90%RH
Operating pressure	under 1.2Mpa
Store temperature	-25°C~+80°C

### LAMPIRAN 3

Data sheet sensor ultrasonik HC – SR04



- Working Voltage DC 5 V
- Working Current 15mA
- Working Frequency 40Hz
- Max Range 4m
- Min Range 2cm
- MeasuringAngle 15 degree
- Trigger Input Signal 10uS TTL pulse
- Echo Output Signal Input TTL lever signal and the range in proportion
- Dimension 45\*20\*15mm

## LAMPIRAN 4

### Koding Raspberry pi tampilan aplikasi

```
import React, { useEffect, useState, useRef, useCallba ck }
from "react";
import Head from "next/head";
import { useRouter } from "next/router";
import appConfig from "../app.json";
import { withApollo } from "../libs/apollo";
import FadeImage from "../components/FadeImage";
import { FontAwesomeIcon } from "@fortawesome/react-
fontawesome";
import { useNotification } from "../components/Notification";
import dayjs from "dayjs";
import Switch from "react-switch";

const Page = (props) => {
  const router = useRouter();
  const notification = useNotification();
  const [status, setStatus] = useState({
    _id: "CONNECTING",
    message: "Menghubungi alat...",
  });
  // console.log({ status });

  const [latestUpdate, setLatestUpdate] = useState({
    displayed: dayjs().format("DD MMM YYYY HH:mm:ss"),
    date: new Date(),
  });
  const [responding, setResponding] = useState(true);
  // console.log({ responding });
  useEffect(() => {
    let timer = setInterval(() => {
      const diff = dayjs().diff(dayjs(latestUpdate.date),
"second");
      // console.log({ diff, latestUpdate });

      if (diff > 5) {
        setResponding(false);
      } else {
        setResponding(true);
      }
    }
  });
}
```

```

    }, 1000);

    return () => {
        clearInterval(timer);
    };
}, [latestUpdate]);

const [data, setData] = useState({});
// console.log({ data });

const ws = useRef();

useEffect(() => {
    const wsUrl =
        process.env.NODE_ENV === "production" &&
        process.env.WS_PRODUCTION_ENDPOINT
        ? process.env.WS_PRODUCTION_ENDPOINT
        :
`ws://${window.location.hostname}:${process.env.WS_PORT}`;
    console.log("Connecting to", wsUrl);

    ws.current = new WebSocket(wsUrl);

    ws.current.onopen = () => {
        console.log("onopen...");
        setStatus({
            _id: "OPENED",
            message: "Berhasil terhubung ke alat.",
        });
    };

    ws.current.onmessage = (e) => {
        setResponding(true);

        var receivedData = e.data;
        // console.log("onmessage...", receivedData);

        let values = receivedData.split(",");
        if (values.length !== 4) return;

        setData((prevData) => {
            const newData = {
                ...prevData,
                mode: parseInt(values[0]),
                relay: parseInt(values[1]),
                flow: parseFloat(values[2]),
                level: parseFloat(values[3]),
            };
            if (prevData.modeUpdate === undefined) {
                newData.modeUpdate = newData.mode;
            }
            // else {
            //     newData.modeUpdate = prevData.mode;
            // }
        });
    };
}, [data]);

```

```

        if (prevData.relayUpdate === undefined) {
            newData.relayUpdate = newData.relay;
        }
        // else {
        //     newData.relayUpdate = prevData.relay;
        // }
        return newData;
    });
    setLatestUpdate({
        displayed: dayjs().format("DD MMM YYYY HH:mm:ss"),
        date: new Date(),
    });
};

ws.current.onclose = () => {
    console.log("onclose...");
    setStatus({
        _id: "CLOSED",
        message: "Koneksi ke alat terputus.",
    });
};

ws.current.onerror = (e) => {
    console.log("onerror...");
    setStatus({
        _id: "ERROR",
        message: "Error ketika menghubungkan ke alat.",
    });
};

return () => {
    if (!ws.current) return;
    ws.current.close();
};
}, []);

return (
    <div>
        <Head>
            <title>{appConfig.name}</title>
        </Head>

        <div className="max-w-4xl mx-auto py-4 px-4">
            <div className="shadow-lg rounded border border-gray-
200 my-8 p-4">
                <div className="flex justify-between">
                    <a href="/" className="hover:opacity-75">
                        <FontAwesomeIcon icon="arrow-left" /> Keluar
                    </a>
                    <a
                        href="#"
                        onClick={(e) => {
                            window.location.reload();
                        }}
                    >

```

```

        className="hover:opacity-75"
      >
        <FontAwesomeIcon icon="sync" /> Refresh dan
Reconnect
      </a>
    </div>

    <div className="flex flex-row justify-center items-
center border-b border-gray-200 pt-4 pb-4">
      <div className="flex-none">
        <FadeImage
          src={appConfig.logo}
          className="block px-2 md:px-3"
          style={{
            maxWidth: 140,
          }}
        />
      </div>
      <div className="pt-4">
        <div>Status Koneksi: {status._id}</div>
        <div className="text-sm">{status.message}</div>
      </div>
    </div>

    <div>
      <div className="font-bold text-center text-lg pt-
3">
        {appConfig.name}
      </div>
      {!responding && (
        <div className="text-sm text-red-500 text-center
pb-2">
          Not Responding!
        </div>
      )}
      <div className="text-sm text-blue-500 text-center
pb-4">
        Latest Update: {latestUpdate.displayed}
      </div>
    </div>

    <div className="py-4">
      <div className="grid grid-cols-1 md:grid-cols-2
md:space-x-8">
        <div>
          <div className="flex justify-between items-
center border-b border-gray-200 py-4">
            <div>
              Mode -{" "}
              <span className="font-bold">
                {data?.mode === 0 ? "MANUAL" : "AUTO"}
              </span>
            </div>
          </div>
          <Switch

```



```

        checked={data?.modeUpdate === 1}
        onChange={e => {
            const modeUpdate = data?.modeUpdate ===
1 ? 0 : 1;
            if (ws.current) {
ws.current.send(`${modeUpdate},${data.relay}`);
                setData({
                    ...data,
                    modeUpdate,
                });
            }
        }}
    />
</div>

<div className="flex justify-between items-
center border-b border-gray-200 py-4">
    <div>
        Status Pompa -{" "}
        <span className="font-bold">
            {data?.relay === 0 ? "TIDAK AKTIF" :
"AKTIF"}
        </span>
    </div>
    {data?.mode === 0 && (
        <Switch
            checked={data?.relayUpdate === 1}
            onChange={e => {
                const relayUpdate = data?.relayUpdate
=== 1 ? 0 : 1;
                // console.log({ relayUpdate, data });
                if (ws.current) {
ws.current.send(`${data.mode},${relayUpdate}`);
                    setData({
                        ...data,
                        relayUpdate,
                    });
                }
            }}
        />
    )}
</div>
</div>

<div>
    <div className="flex justify-between items-
center border-b border-gray-200 py-4">
        <div>
            Water Level -{" "}
            <span className="font-bold">
                {data?.level?.toFixed(1)} %.
            </span>
        </div>
    </div>
</div>

```

```
        </div>
      </div>

      <div className="flex justify-between items-
center border-b border-gray-200 py-4">
        <div>
          Water Flow -{" "}
          <span className="font-bold">
            {data?.flow?.toFixed(1)} mL/s.
          </span>
        </div>
      </div>
    </div>
  </div>
</div>
);
};

export default withApollo({ ssr: true })(Page);
```

## LAMPIRAN 5

### Koding raspberry pi baca sensor

```
require("dotenv").config({
  path: "../.env",
});
const scheduler = require("node-schedule");
const shell = require("shelljs");
const dayjs = require("dayjs");

const dotenv = require("dotenv");
const fs = require("fs");
const envConfig = dotenv.parse(fs.readFileSync("../.env"));
// console.log({ envConfig });
for (const k in envConfig) {
  process.env[k] = envConfig[k];
}

const MAX_CAPACITY_DISTANCE = 15;
const TOP_CAPACITY_THRESHOLD = 80;
const LOW_CAPACITY_THRESHOLD = 20;

let sensorState = {
  mode: 0, // 1 = AUTO, 0 = MANUAL
  relay: 0,
  flow: 0,
  capacity: 0,
};

const start = () => {
  const WebSocket = require("ws");
  const wsEndpointUrl =
    process.env.WS_PRODUCTION_ENDPOINT ||
    `ws://localhost:${process.env.WS_PORT}`;
  console.log("Connect to", wsEndpointUrl);
  const ws = new WebSocket(wsEndpointUrl);

  let running = false;
  const readSensor = async () => {
    if (running) return;
    running = true;

    try {
      if (process.env.NODE_ENV === "production") {
        console.log("Reading flow...");
        var child = shell.exec(`python3 server/03-test-flow-
sensor.py`, {
```

```

        silent: true,
        async: true,
    });
    var result = await new Promise((resolve, reject) => {
        let latestMessage = "";
        child.stdout.on("data", function (data) {
            data = data.trim();
            // console.log({ data })
            latestMessage = data;
            resolve({ stdout: latestMessage });
        });
        child.on("exit", () => {
            resolve({ stdout: latestMessage });
        });
    });
    console.log("> Result", result.stdout);
    const flow = parseFloat(result.stdout.trim());

    console.log("Reading ultrasonic...");
    var child = shell.exec(`python3 server/02-test-
ultrasonic.py`, {
        silent: true,
        async: true,
    });
    var result = await new Promise((resolve, reject) => {
        let latestMessage = "";
        child.stdout.on("data", function (data) {
            data = data.trim();
            // console.log({ data })
            latestMessage = data;
            resolve({ stdout: latestMessage });
        });
        child.on("exit", () => {
            resolve({ stdout: latestMessage });
        });
    });
    console.log("> Result", result.stdout);
    const distance = parseFloat(result.stdout.trim());
    let capacity =
        ((MAX_CAPACITY_DISTANCE - distance) * 100.0) /
MAX_CAPACITY_DISTANCE;
    if (capacity <= 0) capacity = 0;
    else if (capacity >= 100) capacity = 100;

    let relay = sensorState.relay;
    if (sensorState.mode == 1) {
        if (capacity > TOP_CAPACITY_THRESHOLD) {
            let result = shell.exec(
                `python3 server/01-test-relay.py -s INACTIVE`,
                {
                    silent: true,
                }
            );
            relay = 0;
        }
    }

```

```

    } else if (capacity < LOW_CAPACITY_THRESHOLD) {
      let result = shell.exec(
        `python3 server/01-test-relay.py -s ACTIVE`,
        {
          silent: true,
        }
      );
      relay = 1;
    }
  } else {
    if (capacity > TOP_CAPACITY_THRESHOLD) {
      let result = shell.exec(
        `python3 server/01-test-relay.py -s INACTIVE`,
        {
          silent: true,
        }
      );
      relay = 0;
    }
  }
}

console.log("> State", {
  flow,
  capacity,
  relay,
});

sensorState = {
  ...sensorState,
  flow,
  capacity,
  relay,
};
}

let message =
`${sensorState.mode},${sensorState.relay},${sensorState.flow},
${sensorState.capacity}`;
console.log("> Sending", message);
ws.send(message);
console.log(" ");
} catch (err) {}

running = false;
};

ws.on("open", function open() {
  console.log("> On open!");
  const job = scheduler.scheduleJob("* * * * *",
readSensor);
});

ws.on("message", function incoming(data) {
  console.log("> On message", data);

```

```

const values = data.trim().split(",");
console.log({ values });
if (values.length !== 2) return;

const newMode = parseInt(values[0]);
const newRelay = parseInt(values[1]);
console.log({
  newMode,
  newRelay,
});

sensorState.mode = newMode;
sensorState.relay = newRelay;

if (process.env.NODE_ENV === "production") {
  if (sensorState.mode === 0) {
    if (sensorState.capacity > TOP_CAPACITY_THRESHOLD) {
      let result = shell.exec(
        `python3 server/01-test-relay.py -s INACTIVE`,
        {
          // silent: true,
        }
      );
      sensorState.relay = 0;
    } else {
      if (sensorState.relay === 1) {
        let result = shell.exec(
          `python3 server/01-test-relay.py -s ACTIVE`,
          {
            // silent: true,
          }
        );
      } else {
        let result = shell.exec(
          `python3 server/01-test-relay.py -s INACTIVE`,
          {
            // silent: true,
          }
        );
      }
    }
  }
}

let message =
`${sensorState.mode},${sensorState.relay},${sensorState.flow},
${sensorState.capacity}`;
console.log("> Sending", message);
ws.send(message);
console.log(" ");
});

ws.on("close", function close() {

```

```
        console.log("> On close!");
        process.exit();
    });
};

start();
```

## DAFTAR RIWAYAT HIDUP



**MUHAMAD ARYAFATA KUSUMA**, lahir di Malang 02 Maret 2002. Putra pertama dari bapak Hari kusuma dan ibu Siti yunaidah. Mempunyai dua saudara perempuan yang bernama Farachdelia kusuma ningrum dan Anjanie ardhianatayu kusuma. Bertempat tinggal di Ds.sumbertempur Dsn.sumbergirang kecamatan puri Kabupaten Mojokerto. Dengan telah menempuh pendidikan formal:

1. SDI ISLAM PLUS AL AZHAR lulus pada tahun 2014
2. MTS AKSELERASI PP.AMANATUL UMMAH lulus pada tahun 2016
3. MA AKSELERASI PP.AMANATUL UMMAH lulus pada tahun 2018

Pada bulan Agustus 2018 diterima sebagai taruna Politeknik Penerbangan Surabaya. Jurusan Teknik Listrik Bandara angkatan XIII sampai dengan saat ini.