

# Perancangan Monitoring *Coal Handling System* PLTU Kaltim Teluk Berbasis *Internet of Things*

Anshori<sup>1</sup>, Mayda Waruni Kasrani<sup>2</sup>, Khairunnisa Nurhandayani<sup>3</sup>

<sup>1,2,3</sup> Teknik Elektro, Fakultas Teknologi Industri Universitas Balikpapan  
Jln. Pupuk Raya Gn. Bahagia Balikpapan 76114 INDONESIA  
Email: djarumb23@gmail.com

*Abstract- Balikpapan Bay PLTU is a large capacity power plant located in the city of Balikpapan, which is located in East Kalimantan. This PLTU with a capacity of 2X110 MW meets the electricity needs of the Barito Mahakam network, which consists of the East Kalimantan, North Kalimantan, Central Kalimantan and South Kalimantan regions. The main fuel for electricity generation is low grade coal (LRC) or low calorie coal. High Speed Diesel (HSD) oil is used as initial fuel. DCS (Distributed Control System) is used for wood management at PLTU Kaltim Teluk. DCS helps operations by monitoring and controlling the main and supporting components, however, management, supervisors and other staff have difficulty knowing the operating conditions of the generating unit remotely because the monitoring system can only be accessed by the control room operator at the CCR. The aim of this research is to design an Internet of Things-based coal handling system control system for PLTU Kaltim Teluk. This real time based digital monitoring system can collect and display data from every sensor in the field. The research method is a literature study method and a field method to apply the design results and collect monitoring data at the two Balikpapan Bay PLTUs. The results show that testing, which includes integration testing with MySQL, functional testing, and URL and Endpoint testing, was carried out successfully. As expected, all the main features and functions of the PLTU monitoring system are running well. This system has average percentage error 0.1%. So this study has an accuracy rate of 99.9%. This system has the ability to make it easier for managers to monitor the system.*

*Intisari- PLTU Teluk Balikpapan adalah pembangkit listrik dengan kapasitas besar yang terletak di kota Balikpapan, yang terletak di Kalimantan Timur. PLTU dengan kapasitas 2X110 MW ini memenuhi kebutuhan listrik jaringan Barito Mahakam, yang terdiri dari wilayah Kalimantan Timur, Kalimantan Utara, Kalimantan Tengah, dan Kalimantan Selatan. Bahan bakar utama untuk pembangkit listrik adalah batu bara kelas rendah (LRC) atau batubara berkalori rendah. Minyak diesel High Speed (HSD) digunakan sebagai bahan bakar awal. DCS (Sistem Kontrol Terdistribusi) digunakan untuk pengelolaan kayu di PLTU Kaltim Teluk. DCS membantu pengoperasian dengan memantau dan mengontrol komponen utama dan pendukung, namun, manajemen, supervisor, dan staf lainnya kesulitan mengetahui kondisi operasi unit pembangkit dari jarak jauh karena sistem monitoring hanya dapat diakses oleh operator control room di CCR. Tujuan penelitian ini adalah untuk merancang sistem pengendalian coal handling system berbasis Internet of Things untuk PLTU Kaltim Teluk. Sistem pemantauan digital berbasis real time ini dapat mengumpulkan dan menampilkan data dari setiap sensor di lapangan. Metode penelitian berupa metode studi literatur dan metode lapangan untuk menerapkan hasil rancangan dan mengumpulkan data monitoring pada kedua PLTU Teluk Balikpapan. Hasil menunjukkan bahwa pengujian berhasil, yang mencakup pengujian integrasi dengan MySQL, pengujian fungsi, dan pengujian URL dan Endpoint, dilakukan*

dengan sukses. Sesuai dengan harapan, semua fitur dan fungsi utama sistem pemantauan PLTU berjalan dengan baik. Sistem ini memiliki rata-rata presentase error sebesar 0.1% dan memiliki tingkat akurasi sebesar 99.9%. Sistem ini memiliki kemampuan untuk memudahkan manajer untuk memantau sistem.

**Kata Kunci:** PLTU Balikpapan, *Internet of Things*, *Coal Handling System*, *Monitoring*.

## I. PENDAHULUAN

PLTU Teluk Balikpapan adalah pembangkit listrik dengan kapasitas besar yang terletak di kota Balikpapan, yang terletak di Kalimantan Timur. PLTU dengan kapasitas 2X110 MW ini memenuhi kebutuhan listrik jaringan Barito Mahakam, yang terdiri dari wilayah Kalimantan Timur, Kalimantan Utara, Kalimantan Tengah, dan Kalimantan Selatan. Bahan bakar utama untuk pembangkit listrik adalah batu bara kelas rendah (LRC) atau batubara berkalori rendah. Minyak diesel High Speed (HSD) digunakan sebagai bahan bakar awal.

Sebagian besar PLTU yang menggunakan bahan bakar batubara memiliki sistem penanganan atau penyaluran batubara untuk keperluan operasi, yang biasanya disebut sebagai coal handling system [2] Jumlah batubara antara 10.000 dan 12.000 ton digunakan untuk kebutuhan operasi dua unit per hari. Untuk memenuhi kebutuhan batubara, peralatan sistem pengelolaan batubara harus handal. Coal handling equipment termasuk conveyor, crusher, coal bunker, dan coal feeder. Bunker batu bara adalah tempat terakhir untuk menyimpan bahan bakar sebelum digunakan untuk pembakaran boiler.

DCS (Sistem Kontrol Terdistribusi) digunakan untuk pengelolaan kayu di PLTU Kaltim Teluk. DCS membantu pengoperasian dengan memantau dan mengontrol komponen utama dan pendukung. Saat ini, manajemen, supervisor, dan staf lainnya kesulitan mengetahui kondisi operasi unit pembangkit dari jarak jauh karena sistem monitoring hanya dapat diakses oleh operator control room di CCR.

Oleh karena itu, penulis tertarik untuk merancang sistem pengendalian coal handling system berbasis *Internet of Things* untuk PLTU Kaltim Teluk. Sistem pemantauan digital berbasis real time ini dapat mengumpulkan dan menampilkan data dari setiap sensor di lapangan. Data dapat ditampilkan melalui web atau aplikasi.

## II. TINJAUAN PUSTAKA

### A. Visualisasi Data dari Dataset COVID 19

Visualisasi Data dari Dataset COVID 19 Menggunakan Pemrograman Python ditulis oleh Yanuangga Galahartlambang,

Titik Khotiah, Jumain. Tujuan dari penelitian ini adalah untuk membuat gambar atau visualisasi data berdasarkan penyebaran kasus COVID 19 di Indonesia menggunakan pemrograman Python [1]

**B. Rancang Bangun Aplikasi Dashboard Sebagai Media**

Rancang Bangun Aplikasi *Dashboard* Sebagai Media Monitoring Kinerja Mesin Pembangkit Listrik Tenaga Uap (PLTU) pada PT Cahaya Fajar Kaltim oleh Indra Sugiarto mencakup pencatatan manajemen perawatan yang mencakup cek vibrasi yang berbasis *dashboard* sebagai media monitoring [3]

**C. Perancangan Dashboard Dashboard Monitoring Monitoring Berbasis Berbasis Internet of Internet of Things (IoT)**

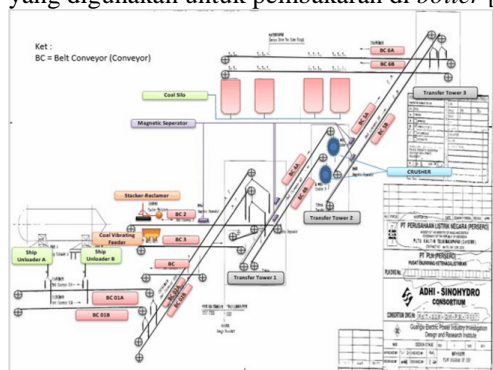
PT Semen Baturaja (Persero) Tbk membuat *Dashboard Monitoring* Berbasis Internet of Things (IoT) yang menggunakan data OPC *server* sebagai input dan menggunakan software Node RED untuk memantau dan memprogram visualnya[4]

**D. 2D Visualization Tools Menggunakan Flask dan AngularJS. Penelitian ini bertujuan**

*2D Visualization Tools* menggunakan *Flask* dan *AngularJS* bertujuan untuk membuat *website* menggunakan *framework Flask* yang berbasis *Python*. Tujuan dari program ini adalah untuk menampilkan data pengguna dari file yang diupload ke sistem[5]

**E. Coal Handling System**

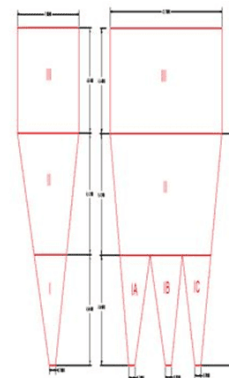
Untuk memastikan bahwa sistem pengelolaan batu bara tetap dalam kondisi yang baik, yaitu bebas dari kerusakan, kecelakaan, dan pembobolan, sistem pengelolaan batu bara menangani berbagai tugas, termasuk pembongkaran batu bara dari kapal atau tongkang (area pengisian), penimbunan atau penyimpanan di area penyimpanan, dan pengisian ke bunker yang digunakan untuk pembakaran di *boiler* [6]



Gambar 1. P&ID Coal Handling System pada PLTU Kaltim Teluk

**F. Coal Bunker**

Coal bunker berfungsi sebagai penampung batubara untuk operasi harian dan memastikan kesiapan unit pembangkit untuk beroperasi sesuai dengan kebutuhan pembebanan [7] Selain itu, *coal bunker* juga dilengkapi dengan pipa masuk CO2, yang akan disemprotkan ke dalam untuk memadamkan kebakaran jika batubara terbakar sendiri.



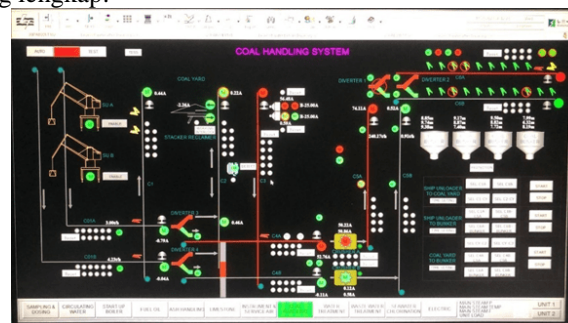
Gambar 2. Bentuk Coal Bunker pada PLTU Kaltim Teluk

**G. Cool Feeder**

*Coal Feeder* berfungsi sebagai penyalur batubara ke *boiler* dan menimbang dan mengontrol aliran massa batubara yang akan masuk ke *boiler*.

**H. Monitoring**

*Monitoring* juga dapat diartikan sebagai proses untuk mengevaluasi apakah tindakan yang dilakukan telah sesuai dengan rencana, menemukan masalah yang perlu ditangani segera, mengevaluasi apakah metode manajemen dan kerja yang digunakan telah tepat untuk mencapai tujuan, dan mengidentifikasi hubungan antara tindakan dan tujuan untuk mengukur kemajuan [8]. Fitur pemantauan ini selalu memberikan informasi terbaru, sehingga semua orang tahu ada perubahan. Selain itu, aplikasi akan ditampilkan di tempat yang sering dilewati atau berkumpul, sehingga aplikasi dapat dipahami oleh semua pihak dan memberikan data hasil unit yang lengkap.

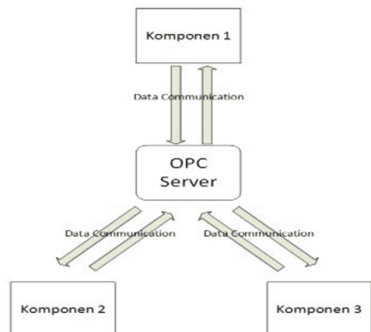


Gambar 3. Gambar Tampilan Monitoring Coal Handling System pada PLTU Kaltim Teluk

**I. OPC**

OPC (*OLE for Process Control*) adalah standar pertukaran data dalam kontrol proses, yang menawarkan metode praktis untuk pertukaran data antara konfigurasi perangkat lunak berbasis sistem operasi *Windows* dan kontrol berbasis lapangan[9]. Tujuan dari standar ini adalah untuk menentukan cara komunikasi data antara sistem otomatisasi waktu nyata dan aplikasi klien yang diinstal pada komputer dengan sistem operasi *Microsoft* dan untuk memungkinkan aplikasi untuk berkomunikasi dengan sistem operasi *Microsoft* [10] OPC menggunakan strategi yang memungkinkan data diakses dari berbagai sumber. Sesuai dengan kebutuhan pengguna OPC, mereka dapat memilih hardware dan *software* yang mereka butuhkan. Setiap perangkat dapat digunakan tanpa khawatir tentang *driver* yang diberikan vendor.

Kerangka kerja OPC memungkinkan PLC berkomunikasi dengan server OPC melalui port serial. Setelah server OPC mendapatkan informasi dari PLC, informasi diubah ke lokasi OPC standar, yang dapat diakses oleh klien OPC. Dengan menggunakan server OPC, klien OPC dapat mempelajari informasi dan mengatur informasi baru ke perangkat.

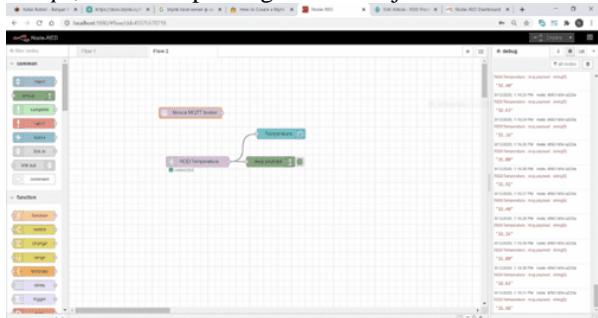


Gambar 4. Skema Komunikasi OPC dengan Komponen Lain[11]

**J. Node RED**

Node RED mengambil rute perangkat lunak alternatif dan menggunakan lingkungan pemrograman visualnya untuk membuat aplikasi *Internet of Things (IoT)*. Pertama, Node RED berfokus pada pengembangan aplikasi sebagai aliran daripada mengkodekan aplikasi sebagai baris kode[12]

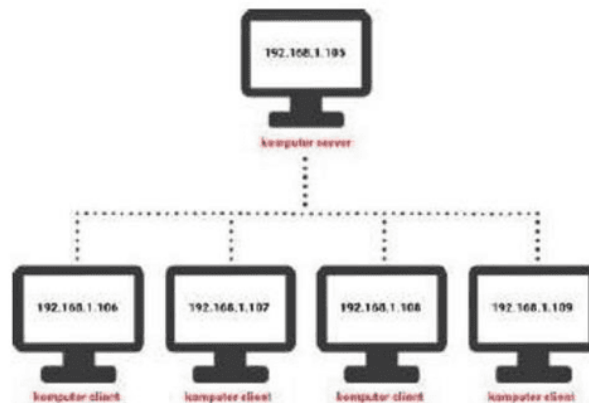
Setiap node dalam proses aliran ini melakukan fungsi tertentu. Meskipun dirancang untuk *Internet of Thing (IoT)*, Node RED masih dapat digunakan untuk berbagai aplikasiumum, karena kesederhanaannya dan kompatibilitasnya dengan berbagai aplikasi. Kemampuan aliran fungsional utama Node RED, yang didasarkan pada bahasa *javascript*, membuat pemrograman menjadi lebih mudah.



Gambar 5. Contoh Flow Diagram pada Nod-RED ( sumber: wiki.rdd-tech.com [13]

**K. PC Server**

Komputer server adalah komputer yang membantu komputer klien dengan berbagai sumber daya. Fungsi server PC termasuk menyediakan sumber daya untuk digunakan bersama, mengelola lalu lintas data, menyimpan file atau dokumen dan data yang dapat diakses bersama dalam jaringan melalui berbagi file, dan mengatur tingkat hak akses dalam jaringan, sehingga tidak semua klien dapat mengakses data yang disimpan di komputer server [4]



Gambar 6. Contoh Koneksi Server Client

**L. Python**

Python adalah bahasa pemrograman komputer yang sebanding dengan C, C++, Pascal, Java, PHP, dan bahasa lainnya [14]. Python adalah bahasa pemrograman interpretatif yang dapat digunakan untuk berbagai tujuan, dan filosofi perancangannya berpusat pada tingkat keterbacaan kode. Python dapat digunakan sebagai bahasa pemrograman dinamis dengan manajemen memori otomatis. Python biasanya digunakan sebagai bahasa skrip, seperti halnya bahasa pemrograman dinamis lainnya. Namun, penggunaan bahasa ini lebih luas, mencakup aplikasi yang biasanya tidak dapat dilakukan dengan bahasa skrip. Python adalah bahasa yang dapat digunakan untuk memenuhi berbagai kebutuhan pengembangan perangkat lunak dan berjalan pada berbagai platform sistem operasi.



Gambar 7. Program Python ( sumber: <https://www.-logique.co.id>[15]

**M. Visual Studio Code ( VS Code)**

Visual Studio Code, juga dikenal sebagai VS Code, adalah sebuah teks editor yang mudah digunakan yang dirancang oleh Microsoft untuk berbagai sistem operasi. Dukungan untuk debugging, penyorotan sintaksis, penyelesaian kode cerdas, snippet, dan refactoring kode termasuk dalam kategori ini [16] Selain itu, teks editor VS Code bersifat *open source*, sehingga Anda dapat melihat kode sumber dan berkontribusi pada pengembangannya. Selain itu, kode sumber dari VS Code ini dapat diakses melalui link Github. Karena mereka dapat berpartisipasi dalam proses pengembangan VS Code di masa mendatang, hal ini membuat VS Code sangat disukai oleh pengembang aplikasi.



Gambar 8. *software Visual Studio Code*  
( sumber: <https://www.niagahoster.co.id> [17] )

**N. Internet of Things (IOT)**

*Internet of Things (IoT)* adalah teknologi yang memungkinkan semua benda di sekitar kita terhubung ke Internet untuk membuat aktivitas sehari-hari lebih mudah dan efektif [18]. Internet adalah jaringan komputer yang saling berhubungan di seluruh dunia yang menggunakan Protokol Standar Internet (TCP/IP) untuk melayani jutaan pengguna di seluruh dunia. *Internet of Things (IoT)* terdiri dari tiga komponen utama: objek fisik atau virtual yang telah terintegrasi ke dalam modul sensor, terhubung ke internet, dan memiliki pusat data pada *server* yang digunakan untuk menyimpan data atau informasi dari aplikasi.



Gambar 9. Konsep dan cara kerja *IoT*  
( sumber: <https://mobnasesemka.com> [19] )

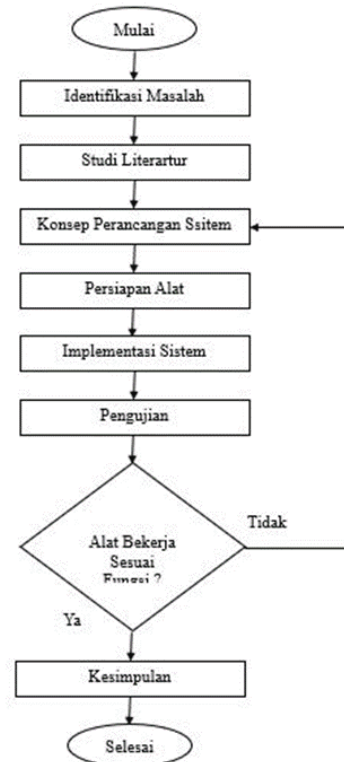
**III. METODE PENELITIAN**

Penelitian ini tentang perancangan sistem pemantauan pengendalian batu bara berbasis *Internet of Things* di PLTU K altim Teluk menggunakan *OPC server* dan *PC server*, yang merupakan aset PLTU Teluk Balikpapan. Bahan penelitian untuk penelitian ini adalah data operasi parameter peralatan. Penelitian ini dimulai pada bulan Maret 2023 sampai dengan bulan Mei 2023.

**A. Diagram Alir Penelitian**

Dengan melihat studi literatur yang telah dikumpulkan, penulis mempelajari konsep dasar dari penelitian yang akan dilakukan dengan instrumen data pendukung, termasuk alat dan peralatan. Diagram menggambarkan proses jalannya penelitian, mulai dari persiapan penelitian, studi literatur, instrumen penelitian, pengumpulan data, dan penyusunan laporan. Tahapan diagram alir penelitian ini dijelaskan dalam diagram berikut: Tujuan penulis adalah untuk mentransmisikan data dari *OPC server* ke *PC server* sehingga data dapat divisualisasikan sebagai data visualisasi web.

Diagram alir penelitian dapat ditunjukkan dalam gambar 10 agar memudahkan pemahaman penelitian.

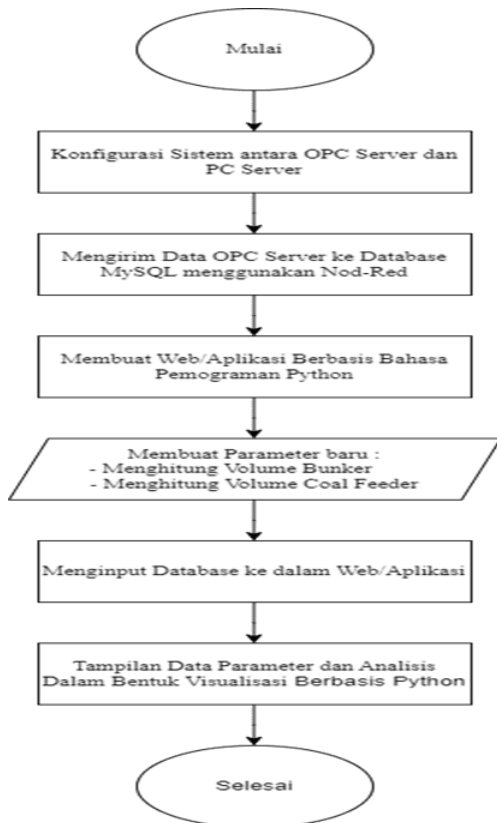


Gambar 10. Diagram Alir Penelitian

**B. Diagram Alir Sistem**

Selanjutnya, dalam diagram alir sistem, konfigurasi komunikasi *OPC server* dengan *PC server* dilakukan. Ini memungkinkan data dikirim dan dibagikan antara kedua komputer tersebut. *Node-RED* berfungsi untuk pengiriman dan penerimaan data; pengaturannya termasuk instalasi dan pembuatan aliran yang diperlukan untuk program visualnya. Selanjutnya, data akan dikirim ke database *MySQL* untuk penyimpanan, dibuat web atau aplikasi dengan bahasa pemrograman python untuk menampilkan parameter data, dan dihitung volume *bunker* dan *feeder coal* menggunakan data parameter dari *OPC server*, menampilkan data *Monitoring Sistem Pengelolaan Batubara* ke visualisasi data yang dapat diakses melalui web atau aplikasi. Diagram alir *system* dapat ditunjukkan dalam gambar 11 agar memudahkan pemahaman penelitian.





Gambar 11. Diagram Alir Sistem

IV. HASIL DAN PEMBAHASAN

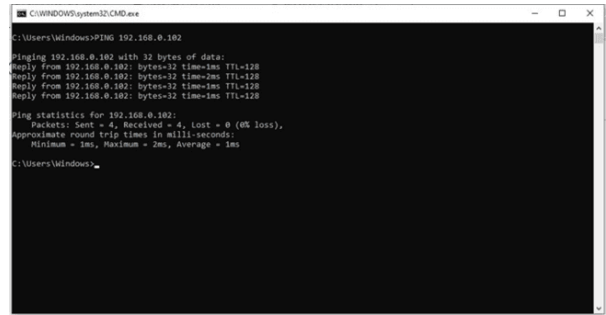
A. Rangkaian Menhubungkan PC Server dan OPC Server

Rangkaian Penghubung PC server dan OPC server menggunakan perangkat jaringan bernama switch untuk berbagi data antara OPC server dan PC server melalui kabel UTP. Switch juga berfungsi untuk menghubungkan antara PC server dan OPC server, dan merupakan perangkat keras yang berguna untuk menghubungkan antar komputer sehingga mereka dapat saling bertukar data (mengirim dan menerima data).



Gambar 12. Rangkaian Penghubung PC Server dan OPC Server

Selanjutnya, lakukan tes. Pengujian koneksi PC server ke OPC server berhasil dilakukan dengan menggunakan perintah PING 192.168.0.102, yang merupakan nomor IP address OPC server. Gambar 13. Pengujian Konektivitas Jaringan PC server ke OPC server menunjukkan hasilnya.



Gambar 13. Pengujian Konektivitas Jaringan PC Server ke OPC Server

B. Rangkaian Menhubungkan PC Server dan Jaringan Internet

Rangkaian ini memantau PC server apakah terhubung ke jaringan internet. Patch panel berfungsi sebagai media penyambung dari kabel FO server ke unit pembangkit di bawah patch panel. Gambar yang menunjukkan bagaimana patch panel bawah disambungkan adalah



Gambar 14. Penyambungan Patch Panel atas ke Patch Panel bawah

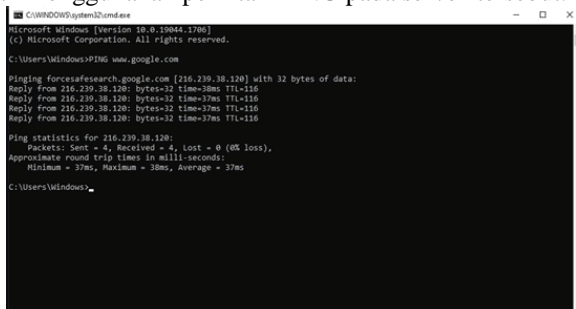
Setelah penyambungan patch panel bawah selesai, perlu menyambungkan patch panel bawah ke media converter. Media converter dapat menyambungkan dua media yang berbeda antara kabel FO dan kabel UTP. Untuk penyambungan ini, komponen yang dibutuhkan adalah media converter, kabel FO, dan kabel UTP, seperti yang ditunjukkan pada Gambar 15.



Gambar 15. Penyambungan Patch Panel bawah ke Media Converter

Pengujian koneksi PC server ke jaringan internet dilakukan setelah penyambungan perangkat keras selesai. Gambar 16 pengujian koneksi PC server ke jaringan internet menunjukkan

hasil pengujian berhasil; server *www.google.com* menanggapi dengan menggunakan perintah PING pada server tersebut.



Gambar 16. Pengujian Koneksi PC Server ke Jaringan Internet

C. Pengujian Node RED

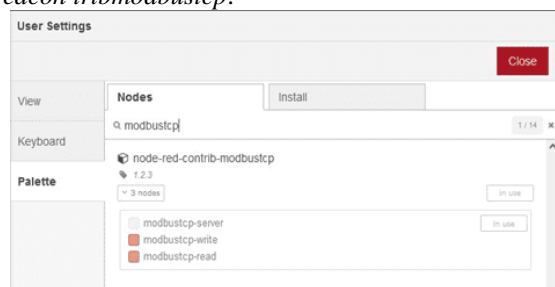
OPC server dapat menerima dan mengirim data dengan pengaturan ini. Sebelum memulai pengujian, Anda harus memulai instalasi pada Node RED. Setelah mengunduh NodeJS, Anda dapat mengecek hasil instalasi. Tampilan instalasi versi NodeJS dapat dilihat pada Gambar 17.



Gambar 17. Tampilan Instalasi Versi NodeJS

Setelah instalasi NodeJS berhasil, instalasi Node RED selanjutnya dilakukan. Tampilan Node RED yang berhasil dijalankan ditunjukkan pada Gambar 18 Setelah berhasil menjalankan Node RED, langkah berikutnya adalah mendapatkan akses ke GUI Grafis Antarmuka Pengguna (GUI), yang merupakan komponen sistem komponen perangkat lunak Node RED dan membantu pengguna berinteraksi dengan perangkat lunak tersebut dengan lebih mudah. Selanjutnya, buka browser dengan alamat *http://localhost:1880*. Setelah server Node RED terbuka, klik tombol di pojok kanan atas dan temukan menu manajemen paleta.

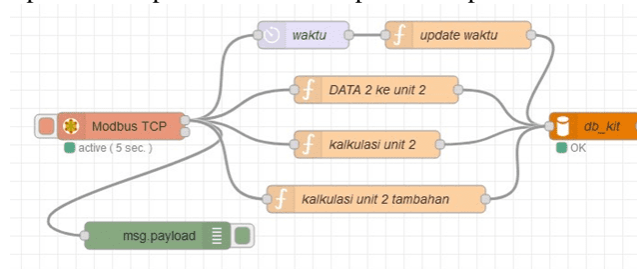
Cari modul yang disebut *modbustcp*, lalu klik tab install pada *node redcon tribmodbustcp*.



Gambar 18. Tampilan Manage Pallete node red contrib modbustcp

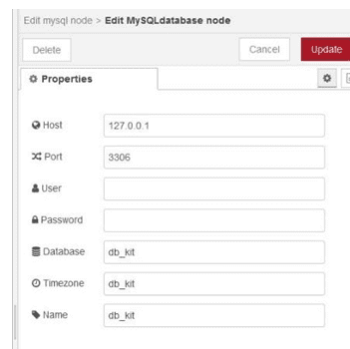
Untuk membuat program Node RED, beberapa node harus digabungkan untuk membuat flow program. Node pertama yang

digunakan adalah *Modbus TCP*, yang dapat membaca data dari OPC server dan mengirim data tersebut ke database MySQL. Tampilan aliran pada Node RED dapat dilihat pada Gambar 19.



Gambar 19. Aliran Kalkulasi Performance pada Node RED

Setelah menyelesaikan pengaturan di sisi klien, langkah berikutnya adalah memilih database. *Node MySQL* berfungsi sebagai *server database* yang akan digunakan dalam penelitian ini. Anda dapat menemukan node ini di manage pallete dengan mencari modul *mysql*, lalu melakukan instalasi pada *node red node mysql*. Pengaturan *database MySQL* dapat dilihat pada Gambar 20.



Gambar 20. Pengaturan Node MySQL pada Node RED

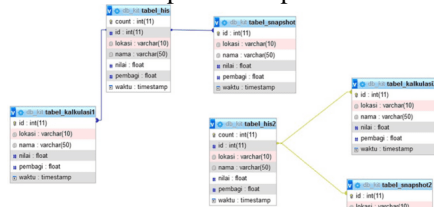
Setelah instalasi dan pengaturan sisi client selesai, pengujian dilakukan untuk memastikan aliran data Node RED berjalan dengan baik. Hasil pengujian ini menunjukkan bahwa debug berfungsi sebagai panel keluaran pesan yang menerima dan memproses data, seperti yang ditunjukkan pada Gambar 21 Panel debug mengeluarkan angka yang memungkinkan pembacaan dan pengiriman angka dari OPC server ke database MySQL.



Gambar 21. Pengujian Pengolahan Data pada Node RED

**D. Pengujian Database MySQL**

Dengan pengaturan ini, kumpulan data OPC server dapat disimpan. Sebelum memulai pengujian, Anda harus melakukan instalasi pada MySQL. Langkah pertama adalah mengunduh XAMPP. Setelah instalasi selesai, di panel kontrol XAMPP, lalu harus memilih untuk memulai modul Apache dan MySQL. Setelah database MySQL berjalan, dilakukan pengujian apakah data dari Node RED dapat dikirim. Hasilnya menunjukkan bahwa data di database MySQL dapat dibaca. Tampilan relasi dari setiap tabel database dapat dilihat pada Gambar 22.

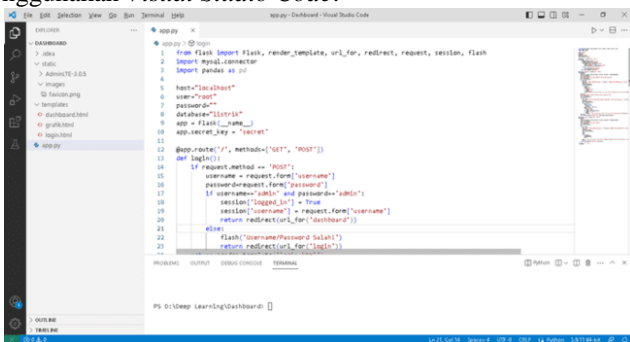


Gambar 22 Tampilan Relasi Tabel Database

Pengujian ini dilakukan untuk memastikan bahwa data sensor dapat dikirim dan disimpan di database MySQL. Ini dilakukan dengan mengakuisisi data dari OPC server, kemudian disimpan di database lokal, dan menampilkan nilai sensor secara visual.

**E. Pengujian Aplikasi Python dan Flask**

Python adalah bahasa pemrograman, dengan Library Flask untuk menyediakan platform Web, dan Code Editor menggunakan Visual Studio Code.



Gambar 23. Tampilan Code Editor di Visual Studio Code

Untuk menjalankan sistem ini, script python berikut digunakan: Bagian program ini memiliki beberapa fungsi, salah satunya adalah sebagai berikut:

• **Segmen Program untuk import library**

```
from flask import Flask,
render_template, url_for,
redirect, request, session, flash
import mysql.connector import
pandas as pd
```

• **Selanjutnya adaah segmen program untuk membaca database MySQL**

```
host="localhost"
user="root"
password=""
database="listrik"
```

• **Segmen Program Login**

```
app = Flask(__name__)
app.secret_key = 'secret'
@app.route('/', methods=['GET',
'POST'])
def login():
    if request.method == 'POST':
        username =
request.form['username']
password=request.form['password']
        if username=='admin' and
password=='admin':
            session['logged_in'] =
True
            session['username'] =
request.form['username']
            return
redirect(url_for('dashboard'))
        else:
            flash('Username/Password
Salah!')
            return
redirect(url_for('login'))
            return
render_template('login.html')
```

• **Segmen Program Tampilan Dashboard**

```
@app.route('/dashboard',
methods=['GET', 'POST'])
def dashboard():
    signal=""
    db =
mysql.connector.connect(host=host,
user=user, password="",
database=database)
    cursor = db.cursor()
    cursor2 =db.cursor()
    q1 = "SELECT DISTINCT
description FROM tes order by
description ASC"
    cursor.execute(q1)
    rsl = cursor.fetchall()
    for r1 in rsl:
        des=r1[0]
        q2="SELECT * FROM tes
WHERE description = '"+str(des)+"'
ORDER BY
STR_TO_DATE(date_rec,'%Y-%m-%d')
DESC LIMIT 1"
```

```

print(q2)
cursor2.execute(q2)
rs2=cursor2.fetchall()
for r2 in rs2:
    val=r2[3]
    tanda="success"
    if(val==0):
        tanda="warning"
    elif(val<0):
        tanda="danger"
    signal+="

```

class='progress-bar bg-"+tanda+"
style='width:
"+str(persen)+"%"></div></div></di
v></div></div>"
        db.close()
        return
    render_template('dashboard.html',s
ignal=signal,progress=progress)

```



Segmen program ini menampilkan dashboard dengan informasi penting.



- Segmen Program menampilkan grafik



```

@app.route('/grafik',
methods=['GET', 'POST'])
def grafik():
    chart = ""
    db =
mysql.connector.connect(host=host,
user=user, password="",
database=database)
    cursor = db.cursor()
    cursor2 = db.cursor()
    q1 = "SELECT DISTINCT
description FROM tes"
    cursor.execute(q1)
    rs1 = cursor.fetchall()
    no=1
    for r1 in rs1:
        des = r1[0]
        q2 = "SELECT * FROM tes
WHERE description =
 '"+str(r1[0])+"' ORDER BY
STR_TO_DATE(date_rec,'%Y-%m-%d %H:
%i:%s') DESC LIMIT 10"
        print(q2)
        cursor2.execute(q2)
        rs2 = cursor2.fetchall()
        cid=1
        val=[]
        id=[]
        for r2 in rs2:
            id.append(cid)
            val.append(r2[3])
            cid+=1

        chart += "<div class='card
ml-1'><div class='card-header bg-
primary'><h3 class='card-title'><i
class='fas fa-chart-line mr-
1'></i><span
class='text'>"+str(r1[0])+"</span>
</h3></div><div class='card-
body'><div class='tab-content p-
0'><div class='chart tab-pane
active' id='revenue-chart'
style='position:relative;height:30
0px'><canvas
id='chart_"+str(no)+"
height='300'
style='height:300px'></canvas></di
v></div></div></div><script>functi

```



Anshori dkk: Perancangan Monitoring Coal Handling System .....



E/P-ISSN: 2549-0842/2528 – 6498


```



```

on random_rgba() {var
a=Math.round,t=Math.random;return'
rgba ('+a (255*t ())+', '+a (255*t ())+'
, '+a (255*t ())+', '+t ().toFixed(1)+'
')}var
speedCanvas=document.getElementById('chart_'+str(no)+'');Chart.default
lts.global.defaultFontFamily='Lato
',Chart.defaults.global.defaultFon
tSize=18;var
speedData={labels:['"+', '".join(m
ap(str,id))+"'],datasets:[{backgro
undColor:random_rgba(),label:'"+str
(r1[0])+"',data:['"+', '".join(ma
p(s
tr,
val))+""]}],chartOptions={legend:{
display:!0,position:'top',labels:{
boxWidth:80,fontColor:'black'}}},l
ineChart=new
Chart(speedCanvas,{type:'line',dat
a:speedData,options:chartOptions})
</script>"
no+=1
return
render_template('grafik.html',char
t=chart)

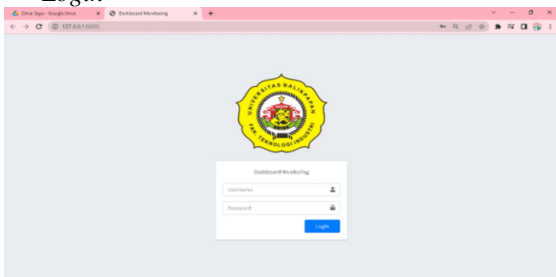
if __name__ == '__main__':
app.run()
    
```

Segmen *script* ini berfungsi untuk menampilkan grafik.

**F. Pengujian Web Dashboard**

Berikut ini adalah hasil tampilan *web dashboard*:

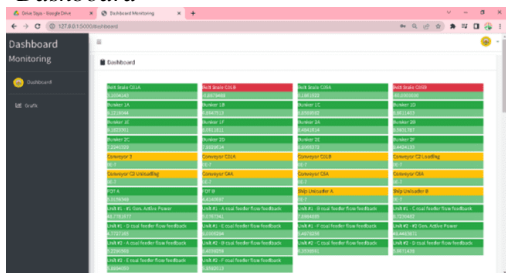
- **Login**



Gambar 24. Tampilan *Login*

Untuk mengakses sistem, tampilan login menggunakan teks untuk memasukkan nama pengguna dan kata sandi. Jika nama pengguna dan kata sandi benar, lalu akan dibawa ke *dashboard*.

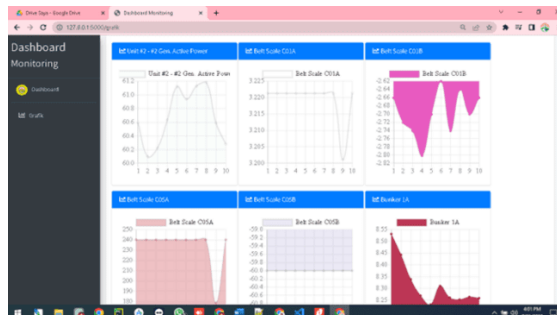
- **Dashboard**



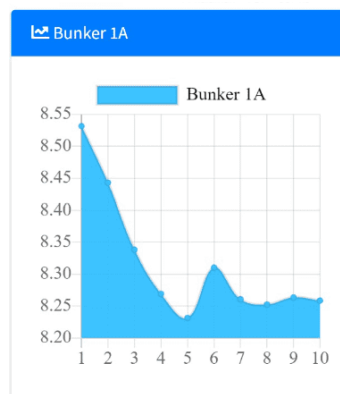
Gambar 25. Tampilan *Dashboard*

Semua informasi penting ditampilkan di halaman *dashboard*; warna hijau menunjukkan bahwa masih aman, warna kuning menunjukkan nilai 0, dan warna merah menunjukkan nilai negatif.

- **Grafik**



Gambar 26. Tampilan Grafik

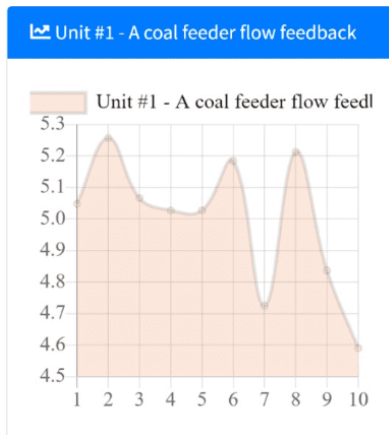


Gambar 27. Grafik tinggi *level Bunker* unit 1A

Tabel 1.1 Hasil pengujian tinggi *level bunker* unit 1A

No	Tanggal	Waktu	Nilai Dashboard (m2)	Nilai DCS (m2)	Selisih	Error (%)
1.	25/04/2023	10.01	8.55	8.54	0.01	0.01
2.	25/04/2023	10.02	8.45	8.45	0.00	0.00
3.	25/04/2023	10.03	8.35	8.34	0.01	0.01
4.	25/04/2023	10.04	8.25	8.27	0.02	0.02
5.	25/04/2023	10.05	8.25	8.23	-0.02	-0.02
6.	25/04/2023	10.06	8.30	8.31	0.01	0.01
7.	25/04/2023	10.07	8.25	8.26	0.01	0.01
8.	25/04/2023	10.08	8.25	8.25	0.00	0.00
9.	25/04/2023	10.09	8.25	8.26	0.01	0.01
10.	25/04/2023	10.10	8.25	8.25	0.00	0.00
						0,1

Tabel diatas menampilkan hasil pengujian tinggi *level bunker* unit 1 yang A yang membandingkan nilai dari *dashboard monitoring* dengan nilai dari DCS (*Distributed Control System*). Dari tabel diatas dapat disimpulkan bahwa dalam 10 kali percobaan dalam 10 menit rata-rata persentase *error* sistem *dashboard monitoring* sebesar 0.1%.



Gambar 28. Grafik jumlah flow pada coal feeder unit 1A

Tabel 4. 1 Hasil pengujian jumlah flow pada coal feeder unit 1A

No	Tanggal	Waktu	Nilai Dashboard (t/h)	Nilai DCS (t/h)	Selisih	Error (%)
1.	25/04/2023	10.01	5.10	5.08	0.02	0.02
2.	25/04/2023	10.02	5.30	5.28	0.02	0.02
3.	25/04/2023	10.03	5.10	5.09	0.01	0.01
4.	25/04/2023	10.04	5.00	5.02	-0.02	-0.02
5.	25/04/2023	10.05	5.00	5.00	0.00	0.00
6.	25/04/2023	10.06	5.20	5.20	0.00	0.00
7.	25/04/2023	10.07	4.70	4.72	-0.02	-0.02
8.	25/04/2023	10.08	5.20	5.20	0.00	0.00
9.	25/04/2023	10.09	4.80	4.82	-0.02	-0.02
10.	25/04/2023	10.10	4.60	4.60	0.00	0.00
						0,1

Tabel diatas menampilkan hasil pengujian jumlah flow pada coal feeder unit 1 yang A yang membandingkan nilai dari dashboard monitoring dengan nilai dari DCS (Distributed Control System). Dari tabel diatas dapat disimpulkan bahwa dalam 10 kali percobaan dalam 10 menit rata-rata persentase error sistem dashboard monitoring sebesar 0.1%. Tampilan data grafik membuat pemantauan sistem lebih mudah karena menampilkan data dalam bentuk grafik.

G. Pengujian Hasil Monitoring

Untuk memastikan bahwa sistem beroperasi dengan benar sesuai dengan spesifikasi fungsional yang telah ditetapkan sebelumnya, pengujian menggunakan blackbox (pengujian fungsional) dilakukan pada sistem monitoring yang menggunakan web framework Python Flask dan database server MySQL. Dalam pengujian blackbox, berikut adalah beberapa prosedur umum:

- Untuk menghindari kerusakan sistem produksi, persiapan lingkungan pengujian dilakukan secara terpisah dari lingkungan produksi.
- Konfigurasi dan instalasi Python Flask dan MySQL sesuai kebutuhan aplikasi.
- Menyiapkan dataset hasil sistem pengawasan mysql sesuai dengan skenario pengujian yang telah ditetapkan sebelumnya.
- Uji setiap URL dan endpoint dalam aplikasi Flask untuk memastikan bahwa permintaan (request)

diteruskan ke fungsi yang benar dan mengembalikan respons (response) yang diharapkan.

- Uji fungsi aplikasi Flask saat ini dan pastikan hasilnya sesuai dengan yang diharapkan.
- Pengujian integrasi dengan MySQL untuk memastikan bahwa terhubung dengan baik untuk melakukan operasi database. Uji query juga digunakan oleh aplikasi untuk memastikan bahwa data disimpan, diambil, dan dihapus dengan benar.
- Pengujian sistem untuk memastikan bahwa sistem bertindak seperti yang diharapkan dalam berbagai situasi, seperti situasi yang tidak terduga atau input yang tidak valid.

V. PENUTUP

A. Kesimpulan

Hasil pengujian sistem pengawasan yang menggunakan Python Flask dan MySQL dapat diambil sebagai berikut:

1. Pengujian berhasil ditandai dengan pengujian, termasuk pengujian URL dan Endpoint, pengujian fungsi, serta pengujian integrasi dengan MySQL, berhasil dilakukan. Semua fitur dan fungsionalitas utama dalam sistem monitoring PLTU berjalan dengan baik sesuai dengan harapan.
2. Pengujian perilaku sistem menunjukkan seberapa baik sistem bertindak dalam berbagai situasi yang diuji. Sistem mampu menyesuaikan dan memberikan respons yang tepat terhadap kondisi yang berbeda. Sistem ini memiliki rata-rata persentase error sebesar 0.1% dan memiliki tingkat akurasi sebesar 99.9%.

B. Saran

Dengan mempertimbangkan temuan di atas, dapat diusulkan sebagai berikut:

1. Menggabungkan fitur cetak dan log untuk memungkinkan pengguna mencetak laporan dan memantau aktivitas sistem.
2. Analisis data historis digunakan untuk melengkapi sistem dengan fitur prediksi dan memberikan gambaran tentang performa di masa depan.
3. Untuk mendukung pengambilan keputusan yang tepat, implementasikan integrasi algoritma prediksi dan tampilkan hasil prediksi dengan tampilan visual yang mudah dipahami.
4. Memastikan fitur prediksi diuji secara menyeluruh dan menerapkan mekanisme peringatan untuk melaporkan potensi masalah dalam sistem pengawasan PLTU.
- 5.

VI. REFERENSI

[1] M. S. Komarudin, Era Agita K, "Analisis Kegagalan Dredging System Pada Outlet Coal," vol. 1, no. 1, pp. 123–138, 2021.

[2] Y. Galahartlambang, T. Khotiah, and J. Jumain, "Visualisasi Data Dari Dataset COVID-19 Menggunakan Pemrograman Python," *J. Ilm. Intech Inf. Technol. J. UMUS*, vol. 3, no. 01, pp. 58–64, 2021, [Online]. Available: <http://jurnal.umus.ac.id/index.php/intech/article/view/417>

[3] I. Sugiarto, "Rancang Bangun Aplikasi Dashboard Sebagai Media Monitoring Kinerja Mesin Pembangkit Listrik Tenaga Uap (PLTU) pada PT Cahaya Fajar Kaltim," skripsi tidak diterbitkan, Program Studi Sistem Informasi, Universitas Institut Bisnis dan Informatika STIKOM, Surabaya, 2019.

- [4] Robby Setiadi, "Perancangan Dashboard Monitoring Berbasis Internet of Things (IoT) di PT Semen Baturaja (Persero) Tbk," skripsi tidak diterbitkan, Program Studi Teknik Elektro, Universitas Jember, Persembahan, 2019.
- [5] J. JEOVANO, "2D Data Visualization Tools Menggunakan Flask dan AngularJS," *J. Intell. Syst. Comput.*, vol. 2, no. 2, pp. 91–97, 2020, doi: 10.52985/insyst.v2i2.184.
- [6] R. Tri Wahyuni, Z. Teknik Elektronika, J. Teknologi Industri, and P. Caltex Riau, "INFOMATEK: Jurnal Informatika, Manajemen dan Teknologi ANALISIS DAMPAK PENERAPAN SISTEM PROTEKSI PLUGGING PADA CHUTE CONVEYOR BERBASIS PLC DI PLTU TENAYAN," vol. 24, pp. 51–58, 2022, doi: 10.23969/infomatek.v24i1.4632.
- [7] D. Nugroho, "Modifikasi Outlet Coal Bunker Dan Outlet Coal Feeder Untuk Mengatasi Plugging Batubara Di Pltu Tanjung Awar-Awar Tuban," *Wahana Tek.*, vol. 06, pp. 48–54, 2017.
- [8] A. Herliana and P. M. Rasyid, "Sistem Informasi Monitoring Pengembangan Software Pada Tahap," *J. Inform.*, no. 1, pp. 41–50, 2016.
- [9] I. D. Pranowo, Y. B. T. Bagastama, and T. A. F. Wibisono, "Communication. between PLC different vendors using OPC server improved with application device," *Telkonnika (Telecommunication Comput. Electron. Control.*, vol. 18, no. 3, pp. 1491–1498, 2020, doi: 10.12928/TELKOMNIKA.v18i3.14757.
- [10] M. Nicola, C.-I. Nicola, M. Duřã, and D. Sacerdotianu, "SCADA Systems Architecture Based on OPC and Web Servers and Integration of Applications for Industrial Process Control," *Int. J. Control Sci. Eng.*, vol. 8, no. 1, pp. 13–21, 2018, doi: 10.5923/j.control.20180801.02.
- [11] U. I. Mujahid, P. I. Siregar, and E. Leksono, "Studi Otomasi Dasar pada Sistem Miniplant Boiler dan Heat Exchanger serta Pengamatan Sistem Melalui HMI dan Database," *J. Otomasi Kontrol dan Instrumentasi*, vol. 9, no. 1, p. 55, 2018, doi: 10.5614/joki.2017.9.1.6.
- [12] S. Mulyono, M. Qomaruddin, and M. Anwar, "Penggunaan Node-RED pada Sistem Monitoring dan Kontrol Green House berbasis Protokol MQTT," *J. Transistor Elektro dan Inform. (TRANSISTOR EI*, vol. 3, no. 1, pp. 31–44, 2018.
- [13] A. Fajrur, "Pengenalan Nod-RED," *wiki.rdd-tech*, 2020. <https://wiki.rdd-tech.com/index.php/knowledge-base/pengenalan-node-red/com> (accessed May 03, 2023).
- [14] Defrianto and F. David, "Visualisasi Data Dalam Bentuk 3 Dimensi Dengan Menggunakan Bahasa Pemrograman Python," *Semin. Nas. Peran. Iptek Menuju Ind. Masa Depan*, pp. 1–6, 2019, doi: 10.21063/PIMIMD5.2019.1.
- [15] F. NKD, "Memahami Pengertian Python dan Mengetahui Kegunaannya," *LOGIQUE*, 2020.
- [16] Agustini and W. J. Kurniawan, "Sistem E-Learning Do'a dan Iqro' dalam Peningkatan Proses Pembelajaran pada TK Amal Ikhlas," *J. Mhs. Apl. Teknol. Komput. dan Inf.*, vol. 1, no. 3, pp. 154–159, 2019, [Online]. Available: <http://www.ejournal.pelitaindonesia.ac.id/JMApTeKsi/index.php/JOM/article/view/526>
- [17] M. Ariffudin, "Menenal Visual Code Studio dan Fitur-Fitur Pentingnya," *NIAGAHOSTERblog*, 2022. <https://www.niagahoster.co.id/blog/visual-code-studio/> (accessed May 21, 2023).
- [18] F. Susannto, N. K. Prasiani, and P. Darmawan, "Implementasi Internet of Things dalam kehidupan sehari-hari," *J. Imagine*, vol. 2, no. 1, pp. 35–40, 2022, doi: 10.35886/imagine.v2i1.329.
- [19] T. Terkini, "Penjelasan dan Cara Kerja Konsep Internet of Things," *mobnasesemka*, 2016. <https://mobnasesemka.com/internet-of-things/> (accessed May 05, 2023).