PERANCANGAN *MONITORING* *OF* *COAL HANDLING SYSTEM* PLTU KALTIM TELUK BERBASIS *INTERNET OF THINGS*

**SKRIPSI**

**Untuk memenuhi salah satu syarat memperoleh gelar**

**Sarjana Teknik pada Program Studi Teknik Elektro**

**Fakultas Teknologi Industri**



Diajukan Oleh:

**Anshori**

**197030717**

**PROGRAM STUDI TEKNIK ELEKTRO**

**FAKULTAS TEKNOLOGI INDUSTRI**

**UNIVERSITAS BALIKPAPAN**

**BALIKPAPAN**

**2024**

# HALAMAN PENGESAHANSKRIPSI

# PERANCANGAN *MONITORING OF* *COAL HANDLING SYSTEM* PLTU KALTIM TELUK BERBASIS *INTERNET OF THINGS*

Dipersiapkan dan disusun oleh

**Anshori**

197030717

Telah dipertahankan di depan dewan penguji

Pada tanggal :

**Susunan Dewan Penguji**

|  |  |
| --- | --- |
| Pembimbing I | Anggota Dewan Penguji I |
|  |  |
| **Mayda Waruni Kasrani, S.T., M.T.**  NIDN. 1101057202 | **Dr. Taqiyuddin, ST. M.Eng.**  NIDN. 111509378303 |
| Pembimbing II | Anggota Dewan Penguji II |
|  |  |
| **K****hairunnisa Nurhandayani, S.T., M.T.**  NIDN. 1129033901 | **Anwar Fattah, ST., M.Ti, P.hd.**  NIDN. 1129077802 |

Skripsi ini telah diterima sebagai salah satu persyaratan

Untuk memperoleh gelar Sarjana

Tanggal :

Ketua Program Studi: S1 Teknik Elektro

**A. Asni B., S.T., M.Eng**

NIK. 006 003 003

Mengetahui,

Dekan Fakultas Teknologi Industri

**Gunawan, S.T., M.T.**

NIP. 197009262005011003

# PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Anshori

NIM : 197030717

Program Studi : Teknik Elektro

Fakultas : Teknologi Industri

menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri bukan merupakan pengambilalihan tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri.

Apabila di kemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Balikpapan,

Yang membuat pernyataan,

Anshori

# LEMBAR ASISTENSI PEMBIMBING

Nama : Anshori

NPM : 197030717

Fakultas : Teknologi Industri

Program Studi : Teknik Elektro

Dosen Pembimbing I : Mayda Waruni Kasrani, S.T., M.T.

Dosen Pembimbing II : Khairunnisa Nurhandayani, S.T., M.T.

# 

# PRAKATA

Puji syukur ke hadirat Allah SWT yang telah melimpahkan rahmat dan barokahnya sehingga penulis dapat menyelesaikan skripsi dengan judul “PERANCANGAN *MONITORING* *OF* *COAL HANDLING SYSTEM* PLTU KALTIM TELUK BEBRBASIS *INTERNET OF THINGS*”. Laporan skripsi ini disusun untuk memenuhi salah satu syarat dalam memperoleh gelar Sarjana Teknik (S.T.) pada Program Studi Teknik Elektro Fakultas Teknologi Industri Universitas Balikpapan.

Dalam melakukan penelitian dan penyusunan laporan skripsi ini penulis telah mendapatkan banyak dukungan dan bantuan dari berbagai pihak. Penulis mengucapkan terima kasih yang tak terhingga kepada:

1. A. Asni B, S.T., M.Eng selaku Ketua Program Studi Teknik Elektro Fakultas Teknologi Industri Universitas Balikpapan yang memberikan izin kepada penulis untuk belajar.

2. Mayda Waruni Kasrani, S. T., M.T.selaku dosen pembimbing I, danKhairunnisa Nurhandayani, S.T., M.T. selaku dosen pembimbing II, yang telah dengan penuh kesabaran dan ketulusan memberikan ilmu dan bimbingan terbaik kepada penulis.

3. Para Dosen Program Studi Teknik Elektro Fakultas Teknologi Industri Universitas Balikpapan yang telah memberikan bekal ilmu kepada penulis.

4. Para Karyawan/wati Program Studi Teknik Elektro Fakultas Teknologi Industri Universitas Balikpapan yang telah membantu penulis dalam proses belajar.

5. Kedua orang tua yang selalu mendoakan penulis dalam menuntut ilmu di Universitas Balikapan.

6. Seluruh karyawan PT. PJB unit kerja PLTU Kaltim Teluk Balikpapan yang memberikan kesempatan untuk menliti.

7. Rekan-rekan Teknik Elektro angkatan 2019 yang telah mendukung agar skripsi selesai.

Penulis menyadari sepenuhnya bahwa laporan skripsi ini masih jauh dari sempurna, untuk itu semua jenis saran, kritik dan masukan yang bersifat membangun sangat penulis harapkan. Akhir kata, semoga tulisan ini dapat memberikan manfaat dan memberikan wawasan tambahan bagi para pembaca dan khususnya bagi penulis sendiri.

Balikpapan,

Anshori

# 

# 

# ABSTRAK

PLTU Kaltim Teluk adalah pembangkit listrik dengan kapasitas besar yang terletak di kota Balikpapan, yang terletak di Kalimantan Timur. PLTU dengan kapasitas 2 × 110 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 *Low Rank Coal* (LRC) atau batubara berkalori rendah. Minyak *High Speed* *Diesel* (HSD) yang digunakan sebagai bahan bakar awal.

*Coal handling system* berfungsi menangani pekerjaan mulai dari pembongkaran batu bara dari kapal tongkang, penimbunan dan penyimpanan di *stock area* ataupun pengisian ke *bunker* yang digunakan untuk pembakaran di *boiler. Distributed Control System (*DCS*)* digunakan untuk membantu pengoperasian, memantau dan mengontrol peralatan - peralatan utama dan pendukung pada *coal handling system*, 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 *Central Control Room*  (CCR).

Tujuan penelitian ini adalah untuk merancang sistem *monitoring* *coal handling system* berbasis *Internet of Things* untuk PLTU Kaltim Teluk. Sistem *monitoring* berbasis digital ini dapat mengumpulkan dan menampilkan data dari setiap sensor di lapangan. Metode penelitian berupa metode studi literatur dan metode di lapangan untuk menerapkan hasil rancangan dan mengumpulkan data *monitoring* pada PLTU Kaltim Teluk. Perancangan *web monitoring* berhasil diimplementasikan dengan menampilkan stastus peralatan dan nilai nilai parameter yang ada di *coal handling system* dengan rata-rata presentase *error* sebesar 0.1% dan memilik tingkat akurasi sebesar 99.9%. dengan tampilan *web* yang dapat diakses oleh semua tingkatan sepeerti manajemen, supervisor maupun staf kantor lainnya.

**Kata kunci:** *IoT, Monitoring, Python, Web*

# ABSTRACT

*PLTU Kaltim Teluk is a large capacity power plant located in the city of Balikpapan, which is located in East Kalimantan. This 2 × 110 MW power plant fulfills the electricity needs of the Barito Mahakam network, which consists of East Kalimantan, North Kalimantan, Central Kalimantan, and South Kalimantan. The main fuel for the power plant is Low Rank Coal (LRC) or low calorie coal. High Speed Diesel (HSD) oil is used as the initial fuel.*

*Coal handling system functions to handle work ranging from unloading coal from barges, stockpiling and storage in the stock area or filling into bunkers used for combustion in boilers. The Distributed Control System (DCS) is used to help operate, monitor and control the main and supporting equipment in the coal handling system, 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 in the Central Control Room (CCR).*

*The purpose of this research is to design an Internet of Things-based coal handling system monitoring system for PLTU Kaltim Teluk. This digital-based monitoring system can collect and display data from each sensor in the field. The research method is in the form of literature study method and field method to implement the design results and collect monitoring data at PLTU Kaltim Teluk. The design of web monitoring was successfully implemented by displaying equipment status and parameter values in the coal handling system with an average percentage error of 0.1% and has an accuracy rate of 99.9%. with a web display that can be accessed by all levels such as management, supervisors and other office staff.*

***Keyword*:** *IoT, Monitoring, Python, Web*

# DAFTAR ISI

[HALAMAN PENGESAHANSKRIPSI …………………………………………...ii](#_Toc158002046)

[PERNYATAAN KEASLIAN TULISAN ………………………………………...iii](#_Toc158002048)

[LEMBAR ASISTENSI PEMBIMBING ………………………………………….iv](#_Toc158002049)

[PRAKATA …………………………………………………………………………v](#_Toc158002050)

[ABSTRAK ……………………………………………………………………….vii](#_Toc158002051)

[ABSTRACT ……………………………………………………………………..viii](#_Toc158002052)

[DAFTAR ISI ……………………………………………………………………...ix](#_Toc158002053)

[DAFTAR GAMBAR ……………………………………………………………..xi](#_Toc158002054)

[DAFTAR TABEL ……………………………………………………………….xiii](#_Toc158002055)

[ARTI LAMBANG DAN SINGKATAN ………………………………………...xiv](#_Toc158002056)

[BAB I PENDAHULUAN …………………………………………………………1](#_Toc158002057)

[1.1 Latar Belakang Masalah …………………………………………………1](#_Toc158002058)

[1.2 Rumusan Masalah ………………………………………………………..2](#_Toc158002059)

[1.3 Tujuan Penelitian ………………………………………………………...2](#_Toc158002060)

[1.4 Batasan Masalah …………………………………………………………3](#_Toc158002061)

[1.5 Manfaat Penelitian ……………………………………………………….3](#_Toc158002062)

[1.6 Sistematika Penulisan ……………………………………………………3](#_Toc158002063)

[BAB II TINJAUAN PUSTAKA …………………………………………………..5](#_Toc158002064)

[2.1 Penelitian yang Relevan …………………………………………………5](#_Toc158002065)

[2.2 Landasan Teori …………………………………………………………...6](#_Toc158002066)

[2.2.1 *Coal Handling System* ……………………………………………………6](#_Toc158002067)

[2.2.2 *Coal Bunker* ………………………………………………………………7](#_Toc158002068)

[2.2.3 *Coal Feeder* ………………………………………………………………8](#_Toc158002069)

[2.2.4 *Monitoring* ……………………………………………………………….8](#_Toc158002070)

[2.2.5 *OPC* ……………………………………………………………………...9](#_Toc158002071)

[2.2.6 *Node-RED* ……………………………………………………………...10](#_Toc158002072)

[2.2.7 *PC Server* ……………………………………………………………….11](#_Toc158002073)

[2.2.8 *Python* ………………………………………………………………….12](#_Toc158002074)

[2.2.9 *Visual Studio Code ( VS Code )* ………………………………………...13](#_Toc158002075)

[2.2.10 *Internet of Things* *(IoT)* ……………………………………………….13](#_Toc158002076)

[BAB III METODE PENELITIAN ………………………………………………15](#_Toc158002077)

[3.1 Waktu dan Tempat Penelitian …………………………………………..15](#_Toc158002078)

[3.2 Metode Pengumpulan Data ……………………………………………..16](#_Toc158002079)

[3.3 Instrumen Penelitian ……………………………………………………18](#_Toc158002080)

[3.4 Jalannya Penelitian ……………………………………………………...18](#_Toc158002081)

[3.5 Metode Analisis Data …………………………………………………..23](#_Toc158002082)

[BAB IV HASIL DAN PEMBAHASAN …………………………………………26](#_Toc158002083)

[4.1 Rangkaian Menghubungkan *PC Server* dan *OPC Server* ………………26](#_Toc158002084)

[4.2 Rangkaian Menghubungkan *PC Server* dan Jaringan Internet …………27](#_Toc158002085)

[4.3 Pengujian *Node-RED* …………………………………………………...29](#_Toc158002086)

[4.4 Pengujian *Database MySQL* …………………………………………..31](#_Toc158002087)

[4.5 Pengujian Aplikasi *Python* dan *Streamlit* ……………………………...32](#_Toc158002088)

[4.6 Pengujian *Web Dashboard* ……………………………………………...33](#_Toc158002089)

[BAB V PENUTUP ……………………………………………………………….49](#_Toc158002090)

[5.1 Kesimpulan ……………………………………………………………….49](#_Toc158002091)

[5.2 Saran ………………………………………………………………………49](#_Toc158002092)

[DAFTAR PUSTAKA …………………………………………………………….50](#_Toc158002093)

[LAMPIRAN …………………………………………………………………….L-1](#_Toc158002094)

# DAFTAR GAMBAR

[Gambar 2. 1 P&ID *Coal handling system* pada PLTU Kaltim Teluk 7](#_Toc158001999)

[Gambar 2. 2 Bentuk *Coal Bunker* pada PLTU Kaltim Teluk 8](#_Toc158002000)

[Gambar 2. 3 Gambar Tampilan *Monitoring Coal Handling* 9](#_Toc158002001)

[Gambar 2. 4 Skema Komunikasi OPC dengan Komponen Lain[12] 10](#_Toc158002002)

[Gambar 2. 5 Contoh *Flow* Diagram pada *Nod-RED* 11](#_Toc158002003)

[Gambar 2. 6 Contoh Koneksi *Server*-*Client* [3] 11](#_Toc158002004)

[Gambar 2. 9 Konsep dan cara kerja *IoT* 14](#_Toc158002005)

[Gambar 3. 1 PLTU Kaltim Teluk Balikpapan 15](#_Toc158001995)

[Gambar 3. 2 Diagram Alir Penelitian 19](#_Toc158001996)

[Gambar 3. 3 Perancangan Sistem Perangkat Keras 22](#_Toc158001997)

[Gambar 3. 4 Diagram Alir Sistem 24](#_Toc158001998)

[Gambar 4. 1 Rangkaian Penghubung *PC* *Server* dan *OPC* *Server* 26](#_Toc158001975)

[Gambar 4. 2 Pengujian Konektivitas Jaringan *PC* *Server* ke *OPC* *Server* 27](#_Toc158001976)

[Gambar 4. 3 Penyambungan *Patch Panel* atas ke *Patch Panel* bawah 27](#_Toc158001977)

[Gambar 4. 4 Penyambungan *Patch Panel* bawah ke Media *Converter* 28](#_Toc158001978)

[Gambar 4. 5 Pengujian Koneksi *PC* *Server* ke Jaringan Internet 28](#_Toc158001979)

[Gambar 4. 6 Tampilan Instalasi Versi *NodeJS* 29](#_Toc158001980)

[Gambar 4. 7 Tampilan *Manage Pallete* *node-red-contrib-modbustcp* 29](#_Toc158001981)

[Gambar 4. 8 Aliran Kalkulasi *Performance* pada *Node-RED* 30](#_Toc158001982)

[Gambar 4. 9 Pengaturan *Node* *MySQL* pada *Node-RED* 30](#_Toc158001983)

[Gambar 4. 10 Pengujian Pengolahan Data pada *Node-RED* 31](#_Toc158001984)

[Gambar 4. 11 Tampilan Relasi Tabel *Database* 32](#_Toc158001985)

[Gambar 4. 12 Tampilan Code Editor di Visual Studio Code 33](#_Toc158001986)

[Gambar 4. 13 Tampilan *Dashboard* 33](#_Toc158001987)

[Gambar 4. 14 Tampilan table *level coal bunker* 34](#_Toc158001988)

[Gambar 4. 15 Tampilan table *coal flow* pada *coal feeder* 35](#_Toc158001989)

[Gambar 4. 16 Tampilan Grafik 36](#_Toc158001990)

[Gambar 4. 17 Grafik tinggi *level Bunker unit* 1A 37](#_Toc158001991)

[Gambar 4. 18 Grafik tinggi *level Bunker unit* 2A 39](#_Toc158001992)

[Gambar 4. 19 Grafik *coal* *flow* pada *coal feeder unit* 1A 42](#_Toc158001993)

[Gambar 4. 20 Grafik *coal* *flow* pada *coal feeder unit* 2A 45](#_Toc158001994)

# 

# DAFTAR TABEL

[Tabel 3. 1 Waktu Kegiatan 16](#_Toc134300757)

[Tabel 3. 2 Komponen Perangkat Keras 21](#_Toc134300758)

[Tabel 3. 3 Komponen Perangkat Lunak 22](#_Toc134300759)

[Tabel 4. 1 Hasil pengujian tinggi *level bunker* unit 1A per 10 menit 37](#_Toc158001781)

[Tabel 4. 2 Hasil pengujian tinggi *level bunker* unit 1A per 30 menit 38](#_Toc158001782)

[Tabel 4. 3 Hasil pengujian tinggi *level bunker* unit 1A per 1 jam 39](#_Toc158001783)

[Tabel 4. 4 Hasil pengujian tinggi *level bunker* unit 2A per 10 menit 40](#_Toc158001784)

[Tabel 4. 5 Hasil pengujian tinggi *level bunker* unit 2A per 30 menit 41](#_Toc158001785)

[Tabel 4. 6 Hasil pengujian tinggi *level bunker* unit 2A per 1 jam 42](#_Toc158001786)

[Tabel 4. 7 Hasil pengujian *coal* *flow* pada *coal feeder*  unit 1A per 10 menit 43](#_Toc158001787)

[Tabel 4. 8 Hasil pengujian *coal* *flow* pada *coal feeder*  unit 1A per 30 menit 44](#_Toc158001788)

[Tabel 4. 9 Hasil pengujian *coal* *flow* pada *coal feeder*  unit 1A per 1 jam 45](#_Toc158001789)

[Tabel 4. 10 Hasil pengujian *coal* *flow* pada *coal feeder*  unit 2A per 10 menit 46](#_Toc158001790)

[Tabel 4. 11 Hasil pengujian *coal* *flow* pada *coal feeder*  unit 2A per 30 menit 47](#_Toc158001791)

[Tabel 4. 12 Hasil pengujian *coal* *flow* pada *coal feeder*  unit 2A per 1 jam 48](#_Toc158001792)

# 

# ARTI LAMBANG DAN SINGKATAN

CCR = *Central Control Room*

DCS = *Distributed Control System*

HMI = *Human Machine Interface*

HSD = *High Speed Diesel*

IoT = *Internet of Things*

IP = *Internet Protocol*

OPC = *Open Platform Communication*

PC = *Personal Computer*

PLC = *Programmable Logic Controller*

PLTU = Pembangkit Listrik Tenaga Uap

TCP = *Transmission Control Protocol*

UI = *User Interface*

UPDK = Unit Pelaksana Pengendalian Pembangkitan

URL = Uniform ResourceLocator

WSGI = *Web Server Gateway Interface*

# BAB I PENDAHULUAN

## 1.1 Latar Belakang Masalah

PLTU Kaltim Teluk merupakan PLTU dengan kapasitas besar yang berada di kota balikpapan, kalimantan timur. PLTU dengan kapasitas 2 × 110 MW ini memasok kebutuhan listrik pada jaringan Barito – Mahakam (Kalimantan Timur, Kalimantan Utara, Kalimantan Tengah, dan Kalimantan Selatan). Dengan Kapasitas 2 × 110 MW kebutuhan akan pasokan batubara yang cukup besar. Batu bara *Low Rank Coal* (LRC) atau batubara berkalori rendah merupakan bahan bakar utama yang digunakan untuk membangkitkan tenaga listrik, dengan bahan bakar awal menggunakan bahan bakar minyak *High Speed Diesel* (HSD).

Secara umum PLTU yang menggunakan bahan bakar batubara mempunyai sistem penanganan atau penyaluran batubara untuk keperluan operasi atau biasa disebut coal handling system [1]. Sebanyak 10.000 – 12.000 ton batubara yang digunakan untuk keperluan operasi 2 unit per harinya. Untuk itu perlunya kehandalan peralatan *coal handling system* agar keperluan batubara tersebut dapat tercapai. Peralatan *coal handling* antara lain *conveyor, crusher, coal bunker dan coal feeder.* *Coal bunker* merupakan tempat penampungan terakhir sebelum digunakan untuk pembakaran boiler.

Sistem monitoring yang digunakan pada *coal handling* PLTU Kaltim Teluk adalah *Distributed Control System* (DCS) yang mendukung pengoperasian seperti pemantauan dan pengontrolan komponen utama dan pendukung. Saat ini sistem *monitoring* hanya dapat diakses oleh operator *control room* di *Central Control Room* (CCR)saja sehingga pihak manajemen membutuhkan informasi untuk memantau kinerja secara efektif dan efisien[2]. Dan juga pada tampilan monitoring *DCS* (*Distributed Control System*) ini masih belum ada tampilan parameter yang bisa memperkirakan berapa lama batu bara yang di dalam *coal bunker* akan habis dan berapa ton jumlah pemakaian batu bara tiap bunker. Sehingga akan memungkinkan batu bara di dalam *coal bunker* kosong dan menyebabkan unit pembangkit *trip*.

Berdasarkan permasalahan tersebut maka diperlukan teknologi informasi yang mampu mentransformasi sistem *monitoring* manual menjadi *system* *monitoring* berbasis digital. Oleh karena itu, penulis tertarik untuk melakukan perancangan *monitoring* *of* *coal handling system* berbasis *Internet of Things* di PLTU Kaltim Teluk. *System monitoring* ini merupakan *system monitoring* berbasisdigital yang dimana dapat mengakuisisi data dari setiap sensor di lapangan dan ditampilkan dalam bentuk yang informatif. Data informasi tersebut dapat ditampilkan dengan berbentuk *Web*.

Dengan *system monitoring* ini, orang-orang di semua tingkatan (dari tingkat operator hingga manajemen) dapat mengakses informasi ini secara *real-time* kapan saja tanpa harus mengunjungi *Central Control Room* (CCR) atau berkoordinasi dengan operator[3]. *Web* *system monitoring* yang dirancang untuk memberikan kemudahan kepada semua pihak terkait performa dan kondisi unit pembangkit.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah disampaikan, maka peneliti dapat merumuskan masalah untuk penelitian ini sebagai berikut :

1. Bagaimana membuat rancangan *monitoring of coal handling system* PLTU Kaltim Teluk sebagai bentuk visualisasi berbasis *Internet of Things*?
2. Bagaimana menambah parameter-parameter baru untuk menambah *performance* *coal handling system* pada PLTU Kaltim Teluk?

## 1.3 Tujuan Penelitian

Adapun tujuan dari penelitian yang dilakukan peneliti sebagai berikut :

1. Merancang *monitoring of coal handling system* berbasis *Internet of Things* agar dapat diakses kapan saja dan di mana saja.
2. Membuat dan menambah parameter-parameter baru untuk *monitoring coal handling system* di PLTU Kaltim Teluk guna meningkatkan *performance* *coal handling system* pada PLTU Kaltim Teluk.

## 1.4 Batasan Masalah

Penelitian yang dilakukan mempunyai batasan-batasan agar terfokus dalam menjawab permasalahan penelitian. Batasan-batasan tersebut adalah sebagai berikut :

1. Studi kasus hanya dilakukan di PLTU Kaltim Teluk.
2. Hanya membahas perancangan *system monitoring* *coal handling* PLTU Kaltim Teluk berbasis *Internet of Things.*
3. Pengujian *security system* yang mendalam tidak akan dilakukan sebagai bagian metodologi penelitian ini.

## 1.5 Manfaat Penelitian

Dari penelitian yang dilakukan diperoleh beberapa manfaat :

1. Dapat menambah wawasan dan pengetahuan tentang membuat sistem *monitoring of coal handling system* berbasis *Internet of Things*
2. Diharapkan dapat memberi manfaat dalam melakukan pemantauan dan pengendalian dalam pengoperasian *coal handling system* di PLTU Kaltim Teluk

## 1.6 Sistematika Penulisan

Bab ini menjelaskan tentang latar belakang yang menjelaskan tentang beberapa alassan yang melandasi untuk melakukan penelitian ini sehingga menghasilkan beberapa rumusan masalah, tujuan penelitian yang diharapkan peneliti, batasan masalah, manfaat yang diharapkan dari penelitian, dan sistematika penulisan penelitian yang dilakukan peneliti.

BAB I PENDAHULUAN

Berisikan latar belakang masalah, perumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian dan sistematika penulisan di dalam penelitian yang dilakukan.

BAB II TINJAUAN PUSTAKA

Berisi tentang penelitian yang relevan, landasan teori, Hipotesis/ Pertanyaan Penelitian (Optional) yang berhubungan dengan penelitian yang akan dilakukan.

BAB III METODE PENELITIAN

Bab ini menjelaskan tentang tahapan-tahapan yang dilakukan dalam penelitian ini. Metode penelitian diawali dengan waktu, tempat dan menjelaskan tentang objek penelitian. Prosedur penelitian yang meliputi penentuan tujuan penelitian, pengumpulan data, perancangan sistem, dan penarikan kesimpulan.

BAB IV HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil-hasil yang diperoleh dan cara pencapaiannya. Uraian harus komprehensif namun tetap ringkas dan padu. Pembahasan hasil penelitian meliputi kelebihan dan kekurangan, termasuk pengujian.

BAB V PENUTUP

Bab ini menjelaskan tentang kesimpulan yang merupakan hasil yang dicapai dan merupakan jawaban rumusan masalah, sedangkan saran bagian ini menguraikan saran-saran yang perlu diperhatikan berdasarkan keterbatasan yang ditemukan dan asumsi yang dibuat, termasuk saran untuk pengembangan lebih lanjut.

# 

# BAB II TINJAUAN PUSTAKA

## 2.1 Penelitian yang Relevan

Penelitian ini menggunakan beberapa jurnal atau penulisan skripsi sebagai sumber pustaka. Penelitian ini ditinjau dari sisi kasus penelitian dan menggunakan pustaka yang relevan.

Pertama, penelitian mengenai visualisasi data dari *dataset* COVID-19 menggunakan Pemrograman *Python* oleh Yanuangga Galahartlambang, Titik Khotiah dan Jumain. Penelitian ini bertujuan untuk menghasilkan visualisasi atau gambaran data berdasarkan penyebaran kasus Covid-19 di Indonesia menggunakan Bahasa pemrograman *python* [4].

Kedua, Rancang Bangun Aplikasi Dashboard Sebagai Media *Monitoring* Kinerja Mesin Pembangkit Listrik Tenaga Uap (PLTU) pada PT Cahaya Fajar Kaltim oleh Indra Sugiarto pencatatan manajemen perawatan meliputi cek vibrasi yang berbasis aplikasi *dashboard* sebagai media *monitoring* [5]*.*

Ketiga, Penelitian yang dilakukan oleh Robby Setiadi, Evelina dan Yudi Wijanarko mengenai Perancangan *Dashboard* *Monitoring* Berbasis *Internet of Things (IoT)* di PT Semen Baturaja (Persero) Tbk. Perancangan ini menggunakan data yang berasal dari *OPC* *server* sebagai nilai *input* dan untuk pemrograman visualnya beserta monitoring ditampilkan menggunakan *software* *Node-RED* [3].

Penelitian yang keempat adalah penelitian yang dilakukan oleh Jerryl Jeovanon yang berjudul 2D *Visualization Tools* Menggunakan *Flask* dan AngularJS. Penelitian ini bertujuan membuat *website* menggunakan *framework Flask* yang berbasis pada bahasa pemrograman *python*. dimanaTujuan pembuatan *software* ini untuk memvisualisasikan data *user* dari *file* yang diupload kepada sistem[6].

Perbedaan penelitian yang akan dilaksanakan peneliti dengan keempat penelitian yang relevan tersebut adalah peneliti akan melakukan perancangan *Monitoring Coal Handling System* PLTU Kaltim Teluk berbasis *Internet of Things*

dengan *OPC* *server* sebagai sumber data untuk pemrograman visual nya dan menggunakan *python* untuk memvisualisasi kan data tersebut.

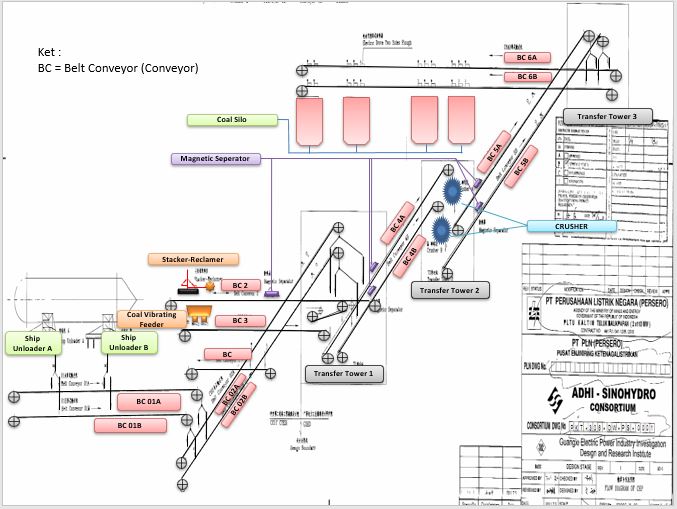
## 2.2 Landasan Teori

Seiring dengan perkembangan zaman dan meningkatnya jumlah penduduk, kebutuhan manusia yang harus dipenuhi juga meningkat termasuk kebutuhan akan energy listrik. Untuk memenuhi kebutuhan energi listrik negara, PLN mendirikan pembangkit listrik di berbagai wilayah dengan beberapa sumber tenaga yaitu air, angin, diesel, uap, surya, panas bumi. Pembangkit Listrik Tenaga Uap (PLTU) Batubara adalah sebuah instalasi pembangkit listrik menggunakan mesin turbin yang diputar oleh uap yang dihasilkan melalui pembakaran batubara yang akan digunakan sebagai bahan bakar untuk pembakaran di *boiler* sebelumnya diproses terlebih dahulu oleh *Coal Handling System*.

## 2.2.1 *Coal Handling System*

*Coal handling system* berfungsi menangani pekerjaan mulai dari pembongkaran batu bara dari kapal/tongkang (*unloading area*), penimbunan/penyimpanan di *stock area* ataupun pengisian ke *bunker* yang digunakan untuk pembakaran di *boiler*[7]. Performa dari *coal handling system* pada industri yang menggunakan batu bara sangatlah berpengaruh pada proses produksi, oleh karena itu keberadaannya perlu dijaga agar tetap dalam kondisi baik yaitu tanpa kerusakan, kecelakaan dan pemborosan.

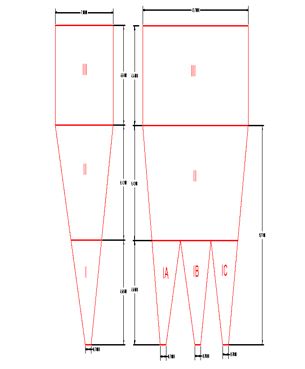
Secara garis besar terdapat beberapa peralatan utama *Coal Handling System* yaitu *belt conveyor*, *stacker reclaimer*, *ship unloader*, *telescopic chute*, *junction tower*, *coal crusher, reclaimer hopper, diverter gate, tripper, scraper conveyor* dan *coal bunker.*



Gambar 2. 1 P&ID *Coal handling system* pada PLTU Kaltim Teluk

## 2.2.2 *Coal Bunker*

*Coal bunker* sebagai penampung batubara untuk operasi harian setiap saat, sehingga kesiapan unit pembangkit untuk bisa beroperasi sesuai kebutuhan pembebanan terjaga dengan baik [8]. selain itu *coal bunker* juga di lengkapi *line* masuknya CO2 jika terjadi kebakaran di dalam *coal bunker* CO2 akan disemprotkan kedalam untuk pemadaman jika batubara terjadi *self combustion.*



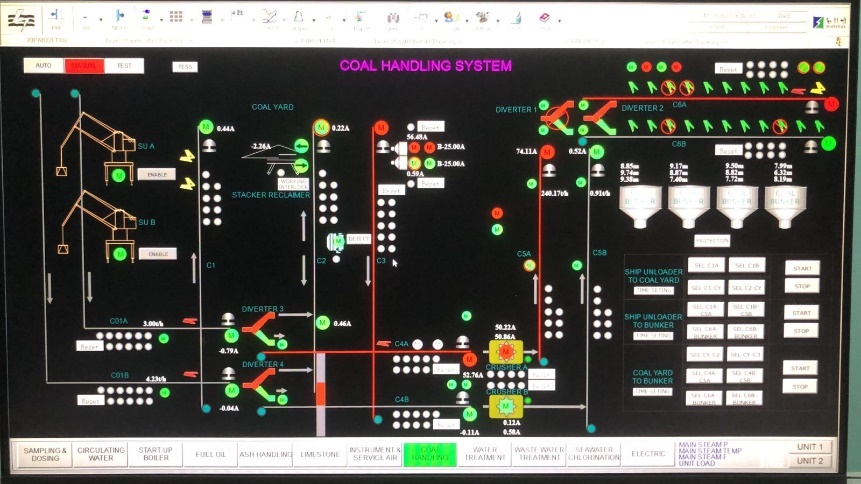
Gambar 2. 2 Bentuk *Coal Bunker* pada PLTU Kaltim Teluk

## 2.2.3 *Coal Feeder*

*Coal feeder* adalah peralatan yang berfungsi untuk menimbang dan mengatur *mass flow rate* batubara yang akan masuk ke *boiler* sekaligus sebagai penyalur batubara ke *boiler.*

## 2.2.4 *Monitoring*

*Monitoring* juga didefinisikan sebagai langkah untuk mengkaji apakah kegiatan yang dilaksanakan telah sesuai dengan rencana, mengidentifikasi masalah yang timbul agar langsung dapat diatasi, melakukan penilaian apakah pola kerja dan manajemen yang digunakan sudah tepat untuk mencapai tujuan, mengetahui kaitan antara kegiatan dengan tujuan untuk memperoleh ukuran kemajuan[9]. Fitur *monitoring* ini memberikan informasi *up-to-date* setiap saat secara terbaru sehingga setiap perubahan dikomunikasikan kepada semua pihak. Selain itu, untuk memudahkan pihak memahami apa yang terjadi di unit, aplikasi akan ditampilkan di tempat-tempat yang sering dilewati atau berkumpul. Dengan aplikasi yang minimal dapat dipahami oleh semua pihak dan memberikan data hasil unit yang lengkap.

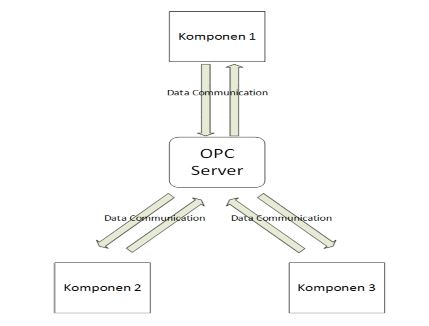


Gambar 2. 3 Gambar Tampilan *Monitoring Coal Handling*

*System* pada PLTU Kaltim Teluk

## 2.2.5 *OPC*

*OLE* *for Process Control* (OPC)adalah pertukaran data standar dalam kontrol proses, yang menyediakan metode praktis untuk pertukaran data antara konfigurasi perangkat lunak berbasis sistem operasi *Windows* dan kontrol berdasarkan kontrol lapangan [10]. Tujuan dari standar ini adalah untuk menentukan metode komunikasi data antara sistem otomatisasi waktu nyata dan aplikasi klien yang diimplementasikan pada komputer dengan sistem operasi *Microsoft* dan untuk memungkinkan aplikasi klien mengakses data otomatisasi dengan cara yang seragam [11]. *OPC* menggunakan metode yang memungkinkan data diakses dari berbagai sumber data yang berbeda. Pengguna *OPC* dapat memilih *software* dan *hardware* sesuai dengan kebutuhannya. Setiap perangkat dapat digunakan tanpa mengkhawatirkan *driver* yang disediakan oleh vendor untuk perangkat tertentu. Kerangka kerja *OPC* adalah bahwa *PLC* berkomunikasi dengan *server* *OPC* melalui *port serial*. Setelah *server* *OPC* memperoleh informasi melalui *port serial*, itu mengubah informasi menjadi lokasi *OPC* standar untuk diakses oleh klien *OPC*. Klien *OPC* dapat mempelajari informasi dan mengatur informasi baru ke perangkat dengan menggunakan *server* *OPC*.

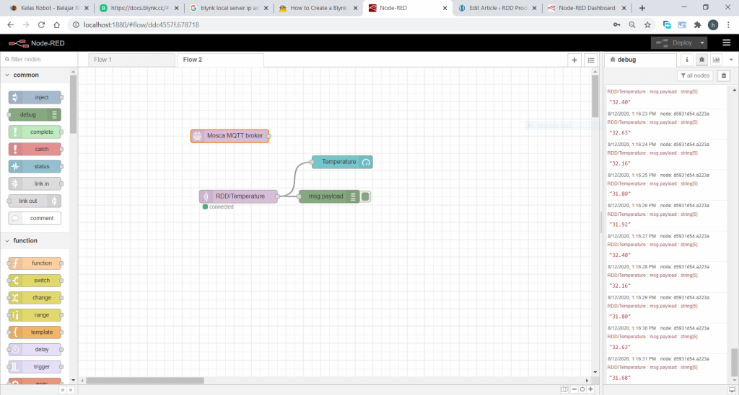


Gambar 2. 4 Skema Komunikasi OPC dengan Komponen Lain[12]

## 2.2.6 *Node-RED*

*Node-RED* adalah sebuah alat berbasis *browser* untuk membuat aplikasi *Internet of Things* (*IoT*) dan lingkungan pemrograman visualnya memungkinkan pengguna untuk dengan mudah membuat aplikasi sebagai aliran. *Node-RED* mengambil rute alternatif perangkat lunak. Pertama, ini adalah bahasa pemrograman visual. Daripada membangun sebuah aplikasi sebagai urutan kode, *Node-RED* berfokus pada program sebagai aliran[13].

Proses aliran ini terdiri dari *node* yang saling berhubungan, di mana setiap *node* melakukan tugas tertentu. Meskipun *Node-RED* dirancang untuk *Internet of Thing* (IoT), *Node-RED* juga dapat digunakan untuk penggunaan umum dan berbagai aplikasi, tetapi penggunaanya terus berkembang karena kesederhanaan dan untuk berbagai aplikasi. Pemrograman melalui *Node-RED* didasarkan pada bahasa *javascript*, yang menciptakan *UI* (*User Interface*) untuk membuat pemrograman lebih mudah menggunakan kemampuan aliran fungsional utama *Node-RED*.

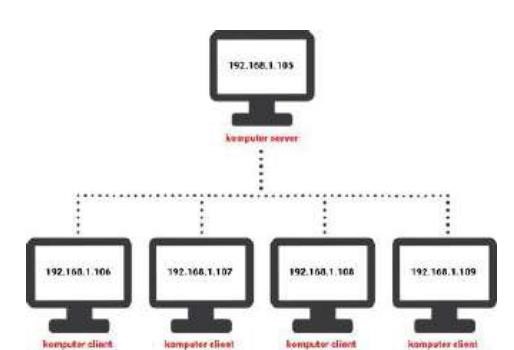


Gambar 2. 5 Contoh *Flow* Diagram pada *Nod-RED*

( sumber: wiki.rdd-tech.com[14] )

## 2.2.7 *PC Server*

*PC* *server* adalah komputer yang melayani komputer klien dengan menyediakan berbagai sumber daya. Fungsi *server* *PC* termasuk menyediakan sumber daya untuk penggunaan bersama, bertanggung jawab untuk mengelola lalu lintas data, dapat menyimpan *file* atau dokumen dan data yang dapat diakses bersama dalam jaringan menggunakan berbagi *file*. Serta dapat mengatur tingkat hak akses dalam jaringan, sehingga tidak semua *client* dapat membuka data yang tersimpan di komputer *server* [3].



Gambar 2. 6 Contoh Koneksi *Server*-*Client* [3]

## 2.2.8 *Python*

*Python* adalah bahasa pemrograman komputer, sama layak nya seperti bahasa C, C*++, Pascal, Java, PHP, Perl,* dan lain-lain [15]. *Python* merupakan bahasa pemrograman interpretatif multiguna dengan filosofi perancangannya yang berfokus pada tingkat keterbacaan kode. Salah satu fitur yang tersedia pada *Python* adalah sebagai bahasa pemograman dinamis yang dilengkapi dengan manajemen memori otomatis. Seperti halnya pada bahasa pemrograman dinamis lainnya, *Python* umumnya digunakan sebagai bahasa skrip meski pada praktiknya penggunaan bahasa ini lebih luas mencakup konteks pemanfaatan yang umumnya tidak dilakukan dengan menggunakan bahasa skrip. *Python* dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan diberbagai platform sistem operasi.

**a*. Python streamlit***

*Python streamlit* adalah *framework* atau *API Python* untuk aplikasi *web*. *Streamlit* juga merupakan aplikasi *web WSGI* yang ringan. Yang berarti *Streamlit* memberi bantuan berupa *tools, library* dan *technology* yang memungkinkan untuk membangun aplikasi web. *Streamlit* merupakan bagian dari kategori kerangka kerja mikro. Kerangka kerja micro biasanya kerangka kerja dengan sedikit atau tidak ada ketergantungan ke *library* eksternal. Kerangka kerja *streamlit* lebih cepat dan mudah dengan kemampuan meningkatkan aplikasi yang kompleks daripada kerangka kerja Django dan juga lebih mudah dipelajari [6].

**b. *Data Visualization***

Visualisasi data adalah penyajian data dalam format gambar atau grafik. Ini memungkinkan para pembuat keputusan untuk melihat analitik yang disajikan secara visual, sehingga dapat memahami konsep-konsep yang sulit atau mengidentifikasi pola-pola baru. Dengan visualisasi interaktif, maka dengan mengambil konsep ini selangkah lebih maju dengan menggunakan teknologi untuk menelusuri bagan dan grafik untuk lebih detail. Tujuan utamanya adalah untuk menyaring dataset besar ke dalam grafik visual untuk memudahkan pemahaman hubungan yang kompleks dalam data [6].

**c.  *AngularJS***

*AngularJS* adalah kerangka kerja struktural untuk aplikasi web dinamis dan open-source. Ini memungkinkan untuk menggunakan *HTML* sebagai bahasa template dan memungkinkan untuk memperluas sintaksis *HTML* untuk mengekspresikan komponen aplikasi dengan jelas dan ringkas. Pengikatan data dan injeksi ketergantungannya menghilangkan banyak kode yang saat ini harus ditulis. Dan itu semua terjadi dalam *browser*, menjadikannya mitra yang ideal dengan teknologi *server* apa pun [6].

## 2.2.9 *Visual Studio Code ( VS Code )*

*Visual Studio Code* (VS *Code*) ini adalah sebuah teks *editor* ringan dan handal yang dibuat oleh *Microsoft* untuk sistem operasi *multiplatform*. . Ini termasuk dukungan untuk *debugging*, kontrol git yang tertanam dan *GitHub*, penyorotan sintaksis, penyelesaian kode cerdas, *snippet*, dan *refactoring* kode [17]. Teks *editor* *VS* *Code* juga bersifat *open source*, yang mana kode sumbernya dapat kalian lihat dan kalian dapat berkontribusi untuk pengembangannya. Kode sumber dari VS *Code* ini pun dapat dilihat di *link Github*. Hal ini juga yang membuat *VS* *Code* menjadi favorit para pengembang aplikasi, karena para pengembang aplikasi bisa ikut serta dalam proses pengembangan *VS* *Code* ke depannya.

### 2.2.10 *Internet of Things* *(IoT)*

*Internet of Things* atau disingkat dengan istilah IoT merupakan teknologi yang menginovasi benda-benda sekitar dengan internet agar aktivitas sehari-hari menjadi lebih mudah dan efisien. *Internet of Things* adalah teknologi yang memungkinkan benda-benda di sekitar kita terhubung dengan Internet [19]. Internet adalah komput global jaringan komputer yang saling berhubungan yang menggunakan *Internet Protocol Standart Suite* (TCP/IP) untuk melayani jutaan pengguna di seluruh dunia. Koneksi jaringan melalui berbagai perangkat elektronik. Konsep *Internet of Things* (IoT) terdiri dari 3 elemen utama yaitu : objek fisik atau berwujud yang telah terintegrasi ke dalam modul sensor, koneksi internet, dan pusat data pada *server* untuk menyimpan data atau informasi dari aplikasi.



Gambar 2. 9 Konsep dan cara kerja *IoT*

( sumber: <https://mobnasesemka>.com[20] )

# BAB III METODE PENELITIAN

## 3.1 Waktu dan Tempat Penelitian

Penelitian perancangan *system* *monitoring* *of coal handling* berbasis *Internet of Things* di PLTU Kaltim Teluk yang dikelola oleh perusahaan PT. PLN Nusantara Power.

Penelitian Perancangan *system* *monitoring* *coal handling* berbasis *Internet of Things*di PLTU Kaltim Teluk ini menggunakan OPC *server* dan PC *server* yang merupakan aset dari PLTU Teluk Balikpapan dimana bahan penelitiannya meliputi data operasi parameter peralatan. Berikut adalah gambar tempat lokasi penelitian pada PT. PLN Nusantara Power, yang dapat dilihat pada gambar 3.1.



Gambar 3. 1 PLTU Kaltim Teluk Balikpapan

Penelitian ini dilakukan pada bulan Maret 2023 sampai dengan bulan Desember 2023 di PT PLN (Persero) UPDK Balikpapan yang terletak di Kelurahan Kariangau, Balikpapan, Kalimantan Timur. Berikut adalah waktu kegiatan untuk jalannya penelitian ini dapat dilihat pada tabel 3.1

Tabel 3. 1 Waktu Kegiatan

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Jenis Kegiatan | Bulan | | | | | | | | | |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 1 | Studi Literatur |  |  |  |  |  |  |  |  |  |  |
| 2 | Pengumpulan Data |  |  |  |  |  |  |  |  |  |  |
| 3 | Pengolahan Data |  |  |  |  |  |  |  |  |  |  |
| 4 | Anaslisa Pengolahan Data |  |  |  |  |  |  |  |  |  |  |
| 6 | Penyusunan Laporan |  |  |  |  |  |  |  |  |  |  |

## 3.2 Metode Pengumpulan Data

Dalam penyusunan penelitian ini penulis mengambil objek penelitian pada PLTU Kaltim Teluk. Pengumpulan data dalam penelitian di PLTU Kaltim Teluk menggunakan 3 cara berikut merupakan uraian yang digunakan :

1. Observasi

Suatu metode pengumpulan data yang digunakan dengan mengamati langsung, melihat dan mengambil suatu data informasi yang dibutuhkan ditempat penelitian ini dilakukan. Peneliti melakukan pengumpulan data yang dilakukan di PLTU Kaltim Teluk.

1. Wawancara

Wawancara merupakan salah satu Teknik pengumpulan data yang dilakukan melalui tatap muka langsung dengan narasumber. Wawancara dilakukan dengan *staff operator* dan *staff* *engineering* pada PLTU Kaltim Teluk yang berhubungan dengan data terkait.

1. Studi Literatur
2. Studi literatur dimana membaca berbagai macam referensi seperti buku-buku,
3. jurnal dan lainnya dan kemudian dipelajari dan dipahami untuk bisa merancang
4. produk tersebu
5. Studi literatur dimana membaca berbagai macam referensi seperti buku-buku,
6. jurnal dan lainnya dan kemudian dipelajari dan dipahami untuk bisa merancang
7. produk tersebu

Studi Literatur dimana membaca berbagai macam refrensi seperti buku-buku, jurnal dan lainnya dan kemudian dipelajari dan dipahami untuk bisa merancang produk tersebut.

Metode pengembangan *software* adalah suatu kerangka kerja yang digunakan untuk menstrukturkan, merencanakan, dan mengendalikan proses pengembangan suatu sistem informasi. metode yang digunakan oleh penulis adalah metode *waterfall.* Metode *Waterfall* merupakan metode yang menyediakan pendekatan alur hidup perangkat lunak secara skuensial atau terurut”. Tahapan dalam metode *waterfall* adalah sebagai berikut :

1. Analisis Kebutuhan *Software*

Tahap ini adalah tahap pengumpulan kebutuhan termasuk dokumen dan *interface* untuk menganalisis/menspesifikasikan kebutuhan perangkat lunak sehingga dapat dipahami kebutuhan *user* guna menentukan solusi *software* yg akan digunakan.

1. Desain

Desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antar muka dan prosedur pengkodean.

1. Implementasi

Desain harus ditranslasikan kedalam program prangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahp desain.

1. Pengujian

Pengujian fokus pada perangkat lunak dari segi logik dan fungsional serta memastikan bahwa semua bagian sudah diuji sehingga keluaran yg dihasilkan sesuai dengan yg diinginkan.

## 

## 3.3 Instrumen Penelitian

Instrumen penelitian yang digunakan penulis dalam melakukan penelitian ini adalah :

1. *OPC* *server*

Sebagai sumber data yang digunakan dalam melakukan penelitian.

1. *PC* *server*

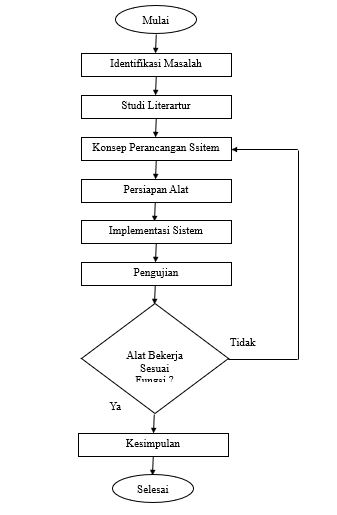
Sebagai komputer yang digunakan untuk mendesain dan menampilkan perancangan sistem.

1. *Laptop*

Sebagai media untuk menyusun laporan penelitian skripsi.

## 3.4 Jalannya Penelitian

Berikut adalah diagram alir penelitian proposal skripsi ini dengan tujuan untuk memahami proses penelitian.



Gambar 3. 2 Diagram Alir Penelitian

Pada Gambar 3.2 menjelaskan tahapan diagram alir penelitian ini yaitu penulis bertujuan cara mentransmisikan data yang diambil dari *OPC* *server* dapat diteruskan ke *PC* *server* sehingga data bisa tampil berupa data visualisasi di *Web*.

1. Penelitian ini dimulai dengan melakukan identifikasi masalah, masalah yang ditemui adalah yaitu :
   1. Membutuhkan informasi unit pembangkit secara *real*.
   2. Informasi yang dibutuhkan digunakan untuk melakukan pemantauan kinerja, operasi dan gangguan dari unit pembangkit.
   3. Sistem kontrol (*Distributed Control System*) DCS hanya dapat diakses oleh operator *control room* di (*Central Control Room*) CCR.
   4. Manajemen dan staf lainnya kesulitan untuk mengetahui kondisi unit pembangkit.

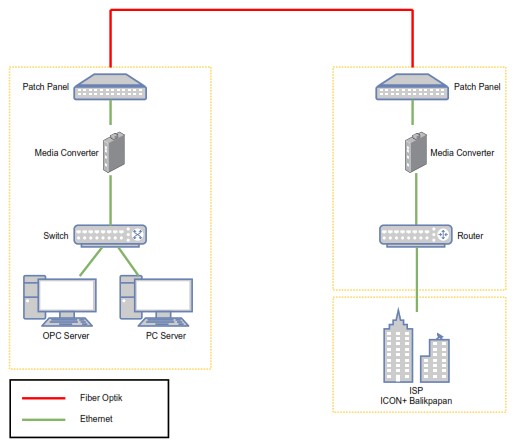
Dari beberapa masalah yang muncul diatas dapat mengambil kesimpulan bahwa diperlukan teknologi informasi yang mampu mentransformasi sistem *monitoring* manual yang awalnya hanya dapat diakses oleh operator *control room* menjadi sistem *monitoring* berbasis digital yang dapat diakses oleh semua kalangan.

1. Melakukan studi literatur untuk mencari informasi yang relevan maupun mencari teori pendukung dalam melaksanakan penelitian perancangan sistem.
2. Setelah melakukan identifikasi masalah dan studi literatur selanjutnya memikirkan konsep perancangan sistem dengan cara menentukan komponen perangkat lunak dan perangkat keras yang dibutuhkan sesuai dengan rancangan yang akan dibuat.
3. Melakukan persiapan alat dan bahan sistem  *monitoring* seperti perakitan dan alur program yang akan digunakan.
   1. Komponen Perangkat Keras
   2. Komponen Perangkat Lunak

Ada beberapa komponen perangkat keras yang digunakan dalam merancang sistem ini, dapat dilihat pada Tabel 3.2 berikut :

Tabel 3. 2 Komponen Perangkat Keras

|  |  |  |  |
| --- | --- | --- | --- |
| 1. | PC *server* | 1 Unit | Sebagai *server* perancangan *system monitoring* |
| 2. | OPC *server* | 1 Unit | Sebagai *data source* dari perancangan *system monitoring* |
| 3. | *Media converter* | 2 Unit | Sebagai penghubung 2 media yang berbeda seperti kabel *FO* dan *UTP* |
| 4. | *Switch* | 1 Unit | Sebagai penyambung dalam jaringan, beroperasi pada *data link layer* |
| 5 | *Patch panel* | 2 Unit | Sebagai media penyambung dari kabel FO ke dalam *server* dengan menggunakan *pigtail* FO |
| 6. | *Router* | 1 Unit | Sebagai perangkat yang dipakai untuk menghubungkan beberapa jaringan, beroperasi pada *network layer* |
| 7. | Kabel *UTP* | 1 Unit, 1 m | Sebagai transmisi data pada sebuah jaringan komputer |
| 8. | Kabel *FO* | 1 Unit, 200 m | Sebagai jaringan telekomunikasi yang menghubungkan perangkat komputer ke jaringan *LAN* |



Gambar 3. 3 Perancangan Sistem Perangkat Keras

Ada beberapa komponen perangkat lunak yang digunakan dalam merancang sistem ini, dapat dilihat pada Tabel 3.3 berikut :

Tabel 3. 3 Komponen Perangkat Lunak

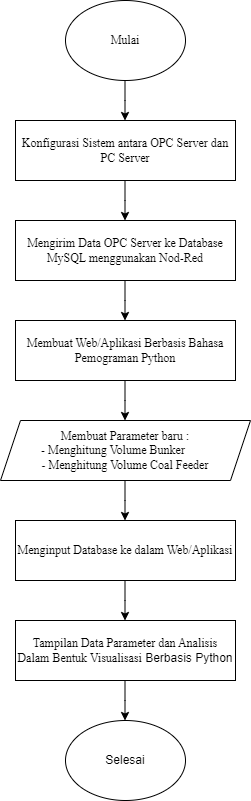
|  |  |  |
| --- | --- | --- |
| **No** | **Nama** | **Fungsi** |
| 1. | *Node-RED* | Sebagai *tool* pemrograman visual berbasis *browser* yang berfungsi untuk menerima dan mengirim data |
| 2. | *MySQL* | Sebagai *database* yang berfungsi menyimpan kumpulan data |
| 3. | *VS Code* | Sebagai perangkat lunak untuk mengedit *source* *code* berbagai [bahasa pemrograman](https://www.niagahoster.co.id/blog/bahasa-pemrograman/" \t "_blank). |

1. Implementasi sistem merupakan bagian penerapan perangkat keras dan perangkat lunak agar alat dapat bekerja sesuai kebutuhan dan fungsinya. Perangkat keras akan dirakit sedikian rupa agar dapat dihubungkan dengan perangkat lunak.
2. Melakukan pengujian dari perancangan sistem yang dibangun.
3. Selanjutnya menarik kesimpulan dari hasil yang diperoleh dari rancangan sistem tersebut.

## 3.5 Metode Analisis Data

Langkah-langkah dalam menganalisis adalah :

1. Melakukan pengujian *hardware* seperti : pengujian koneksi komunikasi antara *OPC* *server* (sebagai komputer *server*) dan *PC* *server* (sebagai komputer *client*).
2. Melakukan pengujian *software* seperti : pengujian *software* *Node-RED*, pengujian *software database* *MySQL,* pengujian *software* Grafana.
3. Melakukan analisis data dari hasil rancangan sistem yang dibuat secara keseluruhan seperti menampilkan data parameter dan analisa data dari sistem *monitoring*.



Gambar 3. 4 Diagram Alir Sistem

1. Melakukan konfigurasi komunikasi *OPC* *server* dengan *PC* *server* sehingga data bisa ditransmisikan dan berbagi antara dua komputer tersebut.
2. Melakukan pengaturan *software* *Node-RED* seperti instalasi dan membuat *flow* yang dibutuhkan untuk program visualnya. *Node-RED* berfungsi untuk pengiriman dan penerimaan data. Data yang diambil selanjutnya akan dikirim ke *database MySQL* untuk penyimpanan data.
3. Melakukan pembuatan *web*/aplikasi dengan Bahasa pemograman *python* untuk menampilkan visualisasi parameter data.
4. Melakukan perhitungan *Volume bunker, volume coal feeder* menggunakan data parameter dari *OPC* *server*.
5. Menampilkan data *Monitoring Coal Handling System* ke tampilan visualisasi data yang berbentuk *web*/aplikasi.
   1. Tambahkan *database* *MySQL* sebagai *data sources*.
   2. Buatlah panel sesuai kebutuhan dan data dapat ditampilkan.

# BAB IV HASIL DAN PEMBAHASAN

Pada bab ini membahas tentang hasil pengujian sistem secara menyeluruh untuk mengetahui sistem dapat berjalan atau tidak sesuai dengan perancangan yang dibuat. Dan juga dapat mengecek setiap peralatan perangkat keras maupun perangkat lunak dapat bekerja sesuai dengan fungsinya atau tidak.

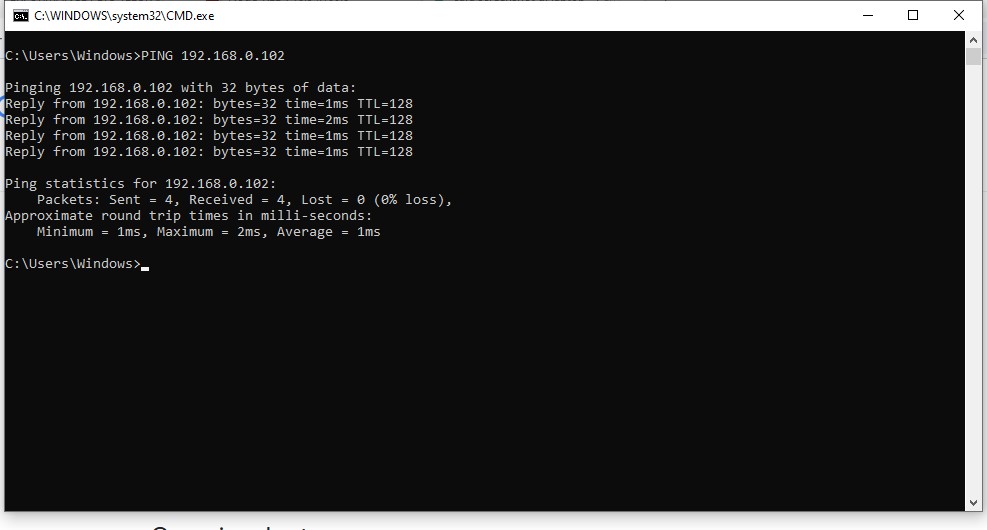
## 4.1 Rangkaian Menghubungkan *PC Server* dan *OPC Server*

Rangkaian Penghubungkan *PC* *server* dan *OPC* *server* ini berfungsi untuk berbagi data dari *OPC* *server* ke *PC* *server*. Dalam rangkaian ini menggunakan perangkat jaringan bernama *switch*. *Switch* berfungsi untuk menghubungkan antara *PC* *server* dan *OPC* *server* juga dibantu dengan kabel *UTP*. *Switch* merupakan perangkat keras yang berguna menghubungkan antar komputer agar bisa saling bertukar data (mengirim dan menerima data).



Gambar 4. 1 Rangkaian Penghubung *PC* *Server* dan *OPC* *Server*

Setelah melakukan rangkaian diatas selanjutnya melakukan pengujian. Hasil pengujian konetivitas dari *PC* *server* ke *OPC* *server* berhasil dilakukan dengan hasil balasan *reply* menggunakan perintah PING 192.168.0.102 dimana nomer *IP* tersebut milik *IP* *address* dari *OPC* *server*, dapat dilihat pada Gambar 4.2 Pengujian Konektivitas Jaringan *PC* *server* ke *OPC* *server*.



Gambar 4. 2 Pengujian Konektivitas Jaringan *PC* *Server* ke *OPC* *Server*

## 4.2 Rangkaian Menghubungkan *PC Server* dan Jaringan Internet

Rangkaian ini berfungsi untuk mengecek *PC* *server* telah terhubung ke jaringan internet atau tidak. Komponen yang dibutuhkan adalah *Patch Panel* sebagai media penyambung dari kabel *FO* *server* menuju ke *patch panel* dibawah (unit pembangkit). Berikut gambar *patch panel* bawah yang telah disambung pada Gambar 4.3 Penyambungan *patch panel* bawah.



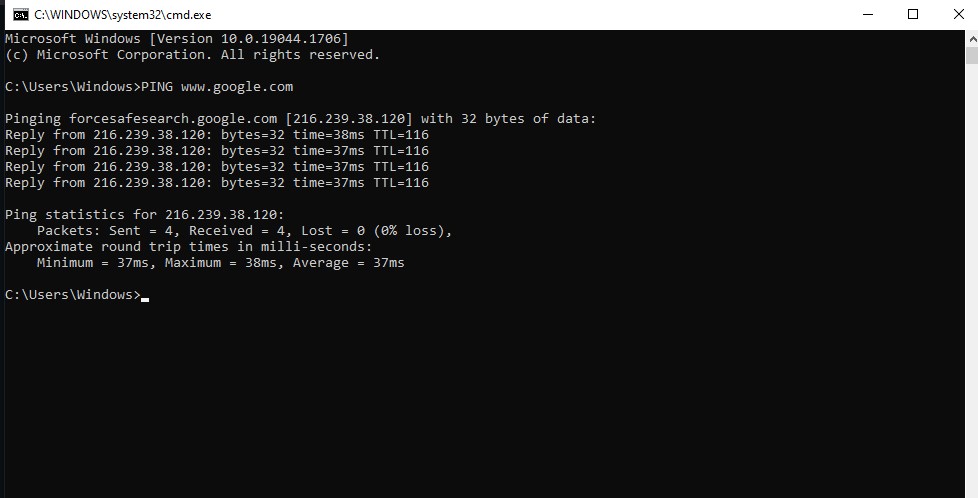
Gambar 4. 3 Penyambungan *Patch Panel* atas ke *Patch Panel* bawah

Setelah melakukan penyambungan *patch panel* bawah selanjutnya melakukan penyambungan *patch panel* bawah ke media *converter*. Media *converter* berfungsi untuk penghubung 2 media yang berbeda antara kabel *FO* dan kabel *UTP*. Komponen yang dibutuhkan dalam penyambungan ini yaitu media *converter*, kabel *FO* dan kabel *UTP* sesuai dengan kebutuhan berikut Gambar 4.4 penyambungan *patch panel* bawah ke media *converter*.



Gambar 4. 4 Penyambungan *Patch Panel* bawah ke Media *Converter*

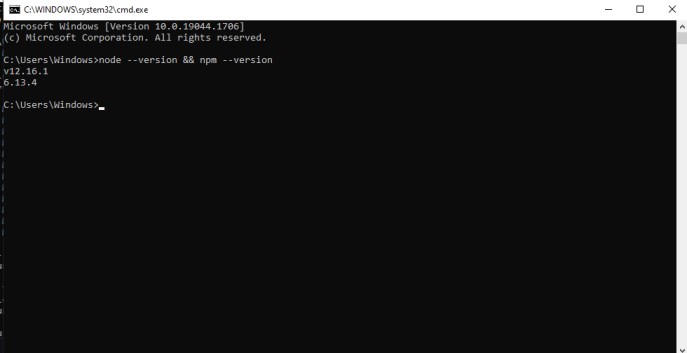
Jika penyambungan perangkat keras telah selesai dilanjut dengan pengujian koneksi *PC* *server* ke jaringan internet. Hasil dari pengujian ini berhasil dapat dilihat bahwa *server* dari www.google.com merespon dengan balasan *reply* dengan menggunakan perintah *PING* pada *server* tersebut. Dapat dilihat pada Gambar 4.5 pengujian koneksi *PC* *server* ke jaringan internet.



Gambar 4. 5 Pengujian Koneksi *PC* *Server* ke Jaringan Internet

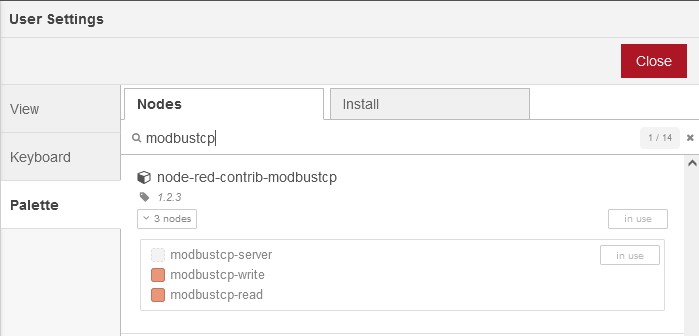
## 4.3 Pengujian *Node-RED*

Pengaturan ini berfungsi untuk menerima dan mengirim data ke alamat mana yang akan dituju oleh *OPC* *server*. Sebelum melakukan pengujian diharuskan untuk melakukan instalasi pada *Node-RED*. Langkah pertama yang perlu dilakukan adalah mengunduh *NodeJS*, setelah berhasil melakukan instalasi kemudian mengecek hasil instalasi. Dapat dilihat pada Gambar 4.6 tampilan instalasi versi *NodeJS*.



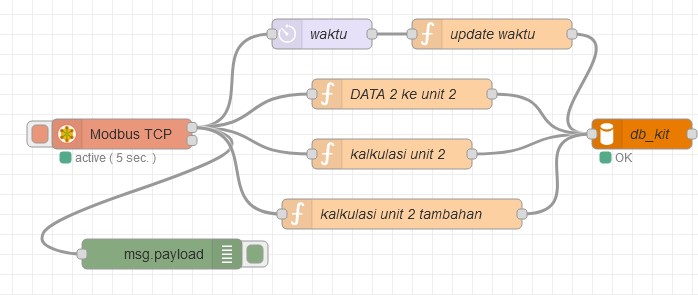
Gambar 4. 6 Tampilan Instalasi Versi *NodeJS*

Setelah berhasil menjalankan *Node-RED* langkah selanjutnya dengan mengakses *GUI* (*Graphical User Interface*) yang merupakan bagian dari sistem komponen perangkat lunak dari *Node-RED* yang berfungsi untuk memudahkan pengguna untuk berinteraksi dengan perangkat lunak tersebut. Kemudian membuka browser dengan alamat *http://localhost:1880*, setelah terbuka *server* *Node-RED* maka klik yang terdapat pada pojok kanan atas dan cari menu *manage pallete*. Carilah modul bernama modbustcp kemudian tab install pada *node-red-contribmodbustcp.*



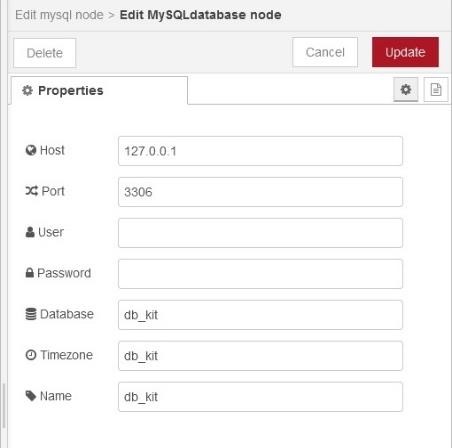
Gambar 4. 7 Tampilan *Manage Pallete* *node-red-contrib-modbustcp*

Dalam membuat program *Node-RED* diperlukan *node*-*node* yang dipakai dan dihubungkan menjadi satu sehingga terbentuk sebuah *flow* program. *Node* pertama yang dipakai adalah *Modbus TCP* merupakan sebuah *node* yang berfungsi untuk membaca data dari *OPC* *server* dan mengirim data *OPC* *server* ke *database* *MySQL*. Dapat dilihat pada Gambar 4.8 tampilan aliran pada *Node-RED*.



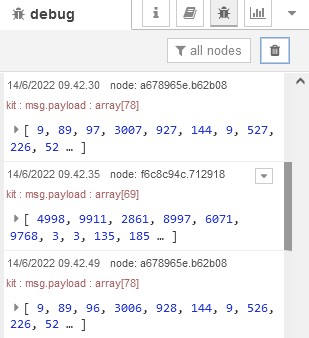
Gambar 4. 8 Aliran Kalkulasi *Performance* pada *Node-RED*

Setelah melakukan pengaturan disisi *client* dilanjutkan dengan memilih *database*. Dengan *node* *MySQL* berfungsi sebagai *server database* yang akan digunakan dalam penelitian ini. *Node* ini dapat ditemukan di *manage pallete* dengan mencari modul mysql kemudian melakukan instalasi pada node-red-node-mysql. Dapat dilihat pada Gambar 4.9 pengaturan database *MySQL*.



Gambar 4. 9 Pengaturan *Node* *MySQL* pada *Node-RED*

Jika instalasi dan pengaturan sisi client sudah dilakukan maka dilanjut dengan pengujian untuk mengetahui berjalannya *flow* data *Node-RED*. Hasil dari pengujian ini berhasil dapat dilihat bahwa *debug* sebagai panel keluaran pesan bahwa data terbaca dan terkirim. Dapat dilihat pada Gambar 4.10 bahwa panel *debug* mengeluarkan angka dimana angka dari *OPC* *server* dapat terbaca dan terkirim ke *database MySQL*.

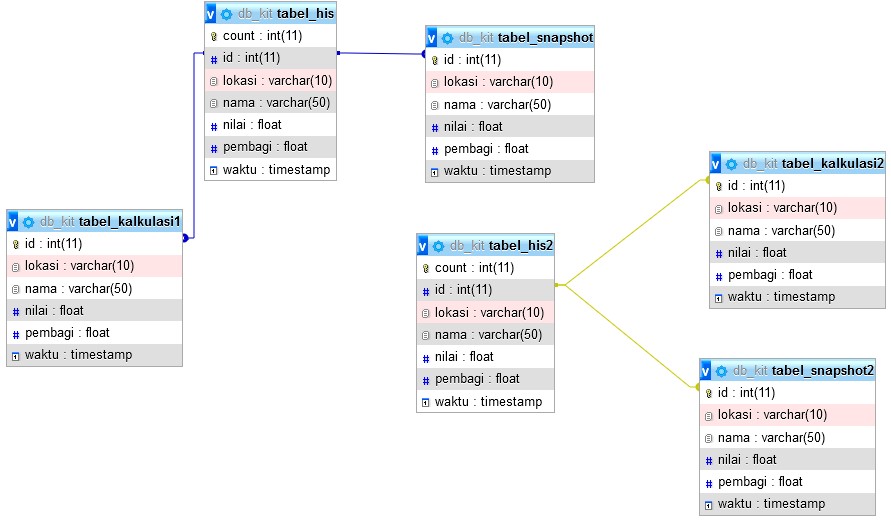


Gambar 4. 10 Pengujian Pengolahan Data pada *Node-RED*

## 4.4 Pengujian *Database MySQL*

Pengaturan ini berfungsi untuk menyimpan kumpulan data dari OPC *server*.

Sebelum melakukan pengujian diharuskan untuk melakukan instalasi pada MySQL. Langkah pertama yang perlu dilakukan adalah mengunduh XAMPP, setelah berhasil melakukan instalasi kemudian klik start pada modul Apache dan MySQL di panel kontrol XAMPP. Setelah *database* MySQL berjalan maka dilanjut dengan melakukan pengujian data dari Node-RED dapat terkirim atau tidak. Hasil pengujian berhasil dengan data terbaca di *database* MySQL. Dapat dilihat pada Gambar 4.11 merupakan tampilan relasi dari setiap tabel *database*.



Gambar 4. 11 Tampilan Relasi Tabel *Database*

Pengujian ini dilakukan untuk memastikan data dari sensor dapat terkirim dan tersimpan di *database* MySQL. Data ini didapat dengan cara mengakuisisi data dari OPC server lalu data tersebut disimpan di database lokal dan menampilkan nilai sensor dalam bentuk visualisasi.

## 4.5 Pengujian Aplikasi *Python* dan *Streamlit*

Bahasa Pemrograman yang digunakan adalah Python, dengan menggunakan *library streamlit* untuk dapat melayani *platform web*, sedangkan *code editor* menggunakan *Visual Studio Code.*

A screenshot of a computer

Description automatically generated

Gambar 4. 12 Tampilan Code Editor di Visual Studio Code

## 4.6Pengujian *Web Dashboard*

Berikut ini adalah hasil tampilan *web dashboard*.

***Dashboard***

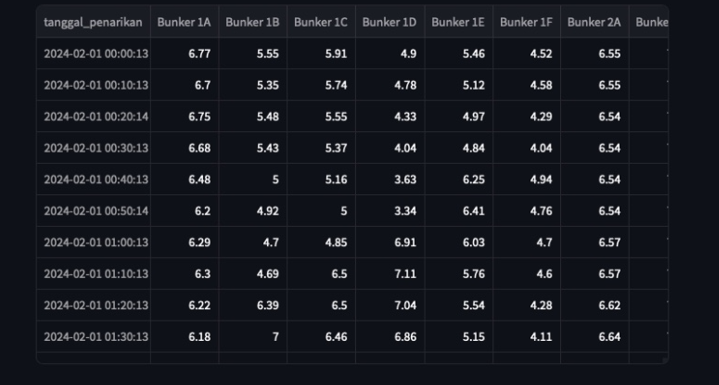
A screenshot of a computer

Description automatically generated

Gambar 4. 13 Tampilan *Dashboard*

*Page dashboard* menampilkan informasi tentang listrik yang dihasilkan dan status perlatan seperti *Conveyor, Ship Unloader, dan Crusher*. Yang berwarna hijau menandakan perlatan sedang beroperasi, dan yang berwarna merah menandakan peralatannya sedang beroperasi.

**Tabel *Level Coal Bunker***



Gambar 4. 14 Tampilan table *level coal bunker*

Tampilan tabel *level coal bunker* berfungsi untuk menampilkan informasi *level* pada setiap *coal bunker* sesuai waktu yang diinginkan.

**Tabel jumlah *flow coal feeder***

****

Gambar 4. 15 Tampilan table *coal flow* pada *coal feeder*

Tampilan tabel *level coal bunker* berfungsi untuk menampilkan informasi *jumlah coal flow* pada setiap *coal feeder* sesuai waktu yang diinginkan.

**Grafik**

A screenshot of a computer

Description automatically generated

Gambar 4. 16 Tampilan Grafik

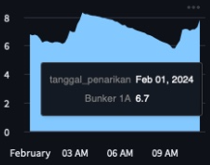
Tampilan data grafik menampilkan data dalam bentuk grafik, dengan demikian akan lebih mudah memantau atau memonitoring system.

Adapun untuk mengetahui persentase akurasi (% *error*) data relatif terhadap OWS (*Operator Work Station*) dari dasbor, maka presentase akurasi dapat dihitung sebagai berikut

Selisih nilai pembacaan (1)

Persentase *error* % = 𝑥 100

Nilai pada DCS



Gambar 4. 17 Grafik tinggi *level Bunker unit* 1A

Tabel 4. 1 Hasil pengujian tinggi *level bunker* unit 1A per 10 menit

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Tanggal** | **Waktu** | **Nilai**  ***Dashboard***  **(m2)** | **Nilai**  **DCS**  **(m2)** | **Selisih** | ***Error* (%)** |
| 1. | 01/02/2024 | 00.00. | 6.77 | 6.77 | 0.00 | 0.00 |
| 2. | 01/02/2024 | 00.10 | 6.7 | 6.74 | 0.04 | 0.05 |
| 3. | 01/02/2024 | 00.20 | 6.75 | 6.75 | 0.00 | 0.01 |
| 4. | 01/02/2024 | 00.30 | 6.68 | 6.68 | 0.00 | 0.02 |
| 5. | 01/02/2024 | 00.40 | 6.48 | 6.48 | 0.00 | 0.00 |
| 6. | 01/02/2024 | 00.50 | 6.2 | 6.24 | 0.04 | 0.05 |
| 7. | 01/02/2024 | 01.00 | 6.29 | 6.29 | 0.00 | 0.00 |
| 8. | 01/02/2024 | 01.10 | 6.3 | 6.32 | 0.02 | 0.03 |
| 9. | 01/02/2024 | 01.20 | 6.22 | 6.22 | 0.00 | 0.00 |
| 10. | 01/02/2024 | 01.30 | 6.18 | 6.18 | 0.00 | 0.00 |
|  |  |  |  |  |  | 0,1 |

Tabel diatas menampilkan hasil pengujian tinggi *level bunker* unit 1 yang A yang membandingan 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%.

Tabel 4. 2 Hasil pengujian tinggi *level bunker* unit 1A per 30 menit

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Tanggal** | **Waktu** | **Nilai**  ***Dashboard***  **(m2)** | **Nilai**  **DCS**  **(m2)** | **Selisih** | ***Error* (%)** |
| 1. | 02/02/2024 | 12.00 | 5.36 | 5.36 | 0.00 | 0.00 |
| 2. | 02/02/2024 | 12.30 | 5.82 | 5.82 | 0.00 | 0.00 |
| 3. | 02/02/2024 | 13.00 | 6.29 | 6.29 | 0.00 | 0.00 |
| 4. | 02/02/2024 | 13.30 | 6.05 | 6.05 | 0.00 | 0.00 |
| 5. | 02/02/2024 | 14.00 | 6.01 | 6.01 | 0.00 | 0.00 |
| 6. | 02/02/2024 | 14.30 | 6.75 | 6.75 | 0.00 | 0.00 |
| 7. | 02/02/2024 | 15.00 | 6.45 | 6.45 | 0.00 | 0.00 |
| 8. | 02/02/2024 | 15.30 | 6.26 | 6.26 | 0.00 | 0.00 |
| 9. | 02/02/2024 | 16.00 | 6.01 | 6.01 | 0.00 | 0.00 |
| 10. | 02/02/2024 | 16.30 | 5.7 | 6.72 | 0.02 | 0.02 |
|  |  |  |  |  |  | 0,1 |

Tabel diatas menampilkan hasil pengujian tinggi *level bunker* unit 1 yang A yang membandingan nilai dari *dashboard monitoring* dengan nilai dari DCS (*Distributed Control System*). Dari tabel diatas dapat disimpulkan bahwa dalam 10 kali percobaan dalam 30 menit rata-rata persentase *error* sistem *dashboard monitoring* sebesar 0.1%.

Tabel 4. 3 Hasil pengujian tinggi *level bunker* unit 1A per 1 jam

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Tanggal** | **Waktu** | **Nilai**  ***Dashboard***  **(m2)** | **Nilai**  **DCS**  **(m2)** | **Selisih** | ***Error* (%)** |
| 1. | 03/02/2024 | 14.00 | 5.58 | 5.58 | 0.00 | 0.00 |
| 2. | 03/02/2024 | 15.00 | 6.93 | 6.93 | 0.00 | 0.00 |
| 3. | 03/02/2024 | 16.00 | 6.46 | 6.46 | 0.00 | 0.00 |
| 4. | 03/02/2024 | 17.00 | 5.66 | 5.66 | 0.00 | 0.00 |
| 5. | 03/02/2024 | 18.00 | 7.16 | 7.16 | 0.00 | 0.00 |
| 6. | 03/02/2024 | 19.00 | 6.93 | 6.93 | 0.00 | 0.00 |
| 7. | 03/02/2024 | 20.00 | 6.33 | 6.33 | 0.00 | 0.00 |
| 8. | 03/02/2024 | 21.00 | 5.41 | 5.41 | 0.00 | 0.00 |
| 9. | 03/02/2024 | 22.00 | 6.07 | 6.07 | 0.00 | 0.00 |
| 10. | 03/02/2024 | 23.00 | 6.57 | 6.57 | 0.00 | 0.00 |
|  |  |  |  |  |  | 0 |

Tabel diatas menampilkan hasil pengujian tinggi *level bunker* unit 1 yang A yang membandingan nilai dari *dashboard monitoring* dengan nilai dari DCS (*Distributed Control System*). Dari tabel diatas dapat disimpulkan bahwa dalam 10 kali percobaan dalam 1 jam rata-rata persentase *error* sistem *dashboard monitoring* sebesar 0%.

A screenshot of a graph

Description automatically generated

Gambar 4. 18 Grafik tinggi *level Bunker unit* 2A

Tabel 4. 4 Hasil pengujian tinggi *level bunker* unit 2A per 10 menit

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Tanggal** | **Waktu** | **Nilai**  ***Dashboard***  **(m2)** | **Nilai**  **DCS**  **(m2)** | **Selisih** | ***Error* (%)** |
| 1. | 01/02/2024 | 00.00. | 6.55 | 6.55 | 0.00 | 0.00 |
| 2. | 01/02/2024 | 00.10 | 6.55 | 6.55 | 0.00 | 0.00 |
| 3. | 01/02/2024 | 00.20 | 6.54 | 6.54 | 0.00 | 0.00 |
| 4. | 01/02/2024 | 00.30 | 6.54 | 6.54 | 0.00 | 0.00 |
| 5. | 01/02/2024 | 00.40 | 6.54 | 6.54 | 0.00 | 0.00 |
| 6. | 01/02/2024 | 00.50 | 6.54 | 6.54 | 0.00 | 0.00 |
| 7. | 01/02/2024 | 01.00 | 6.57 | 6.57 | 0.00 | 0.00 |
| 8. | 01/02/2024 | 01.10 | 6.57 | 6.57 | 0.00 | 0.00 |
| 9. | 01/02/2024 | 01.20 | 6.62 | 6.62 | 0.00 | 0.00 |
| 10. | 01/02/2024 | 01.30 | 6.64 | 6.64 | 0.00 | 0.00 |
|  |  |  |  |  |  | 0 |

Tabel diatas menampilkan hasil pengujian tinggi *level bunker* unit 2 yang A yang membandingan 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 0.0%.

Tabel 4. 5 Hasil pengujian tinggi *level bunker* unit 2A per 30 menit

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Tanggal** | **Waktu** | **Nilai**  ***Dashboard***  **(m2)** | **Nilai**  **DCS**  **(m2)** | **Selisih** | ***Error* (%)** |
| 1. | 02/02/2024 | 12.00 | 6.8 | 6.83 | 0.03 | 0.04 |
| 2. | 02/02/2024 | 12.30 | 6.8 | 6.82 | 0.02 | 0.02 |
| 3. | 02/02/2024 | 13.00 | 6.8 | 6.82 | 0.02 | 0.02 |
| 4. | 02/02/2024 | 13.30 | 6.8 | 6.84 | 0.04 | 0.05 |
| 5. | 02/02/2024 | 14.00 | 6.79 | 6.79 | 0.00 | 0.00 |
| 6. | 02/02/2024 | 14.30 | 6.79 | 6.79 | 0.00 | 0.00 |
| 7. | 02/02/2024 | 15.00 | 6.79 | 6.79 | 0.00 | 0.00 |
| 8. | 02/02/2024 | 15.30 | 6.79 | 6.79 | 0.00 | 0.00 |
| 9. | 02/02/2024 | 16.00 | 6.8 | 6.80 | 0.00 | 0.00 |
| 10. | 02/02/2024 | 16.30 | 6.8 | 6.80 | 0.02 | 0.02 |
|  |  |  |  |  |  | 0,1 |

Tabel diatas menampilkan hasil pengujian tinggi *level bunker* unit 2 yang A yang membandingan nilai dari *dashboard monitoring* dengan nilai dari DCS (*Distributed Control System*). Dari tabel diatas dapat disimpulkan bahwa dalam 10 kali percobaan dalam 30 menit rata-rata persentase *error* sistem *dashboard monitoring* sebesar 0.1%.

Tabel 4. 6 Hasil pengujian tinggi *level bunker* unit 2A per 1 jam

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Tanggal** | **Waktu** | **Nilai**  ***Dashboard***  **(m2)** | **Nilai**  **DCS**  **(m2)** | **Selisih** | ***Error* (%)** |
| 1. | 03/02/2024 | 14.00 | 6.86 | 6.86 | 0.00 | 0.00 |
| 2. | 03/02/2024 | 15.00 | 6.86 | 6.86 | 0.00 | 0.00 |
| 3. | 03/02/2024 | 16.00 | 6.86 | 6.86 | 0.00 | 0.00 |
| 4. | 03/02/2024 | 17.00 | 6.86 | 6.86 | 0.00 | 0.00 |
| 5. | 03/02/2024 | 18.00 | 6.84 | 6.84 | 0.00 | 0.00 |
| 6. | 03/02/2024 | 19.00 | 6.84 | 6.84 | 0.00 | 0.00 |
| 7. | 03/02/2024 | 20.00 | 6.83 | 6.83 | 0.00 | 0.00 |
| 8. | 03/02/2024 | 21.00 | 6.83 | 6.83 | 0.00 | 0.00 |
| 9. | 03/02/2024 | 22.00 | 6.75 | 6.75 | 0.00 | 0.00 |
| 10. | 03/02/2024 | 23.00 | 6.75 | 6.75 | 0.00 | 0.00 |
|  |  |  |  |  |  | 0 |

Tabel diatas menampilkan hasil pengujian tinggi *level bunker* unit 2 yang A yang membandingan nilai dari *dashboard monitoring* dengan nilai dari DCS (*Distributed Control System*). Dari tabel diatas dapat disimpulkan bahwa dalam 10 kali percobaan dalam 1 jam rata-rata persentase *error* sistem *dashboard monitoring* sebesar 0%.

A screenshot of a graph

Description automatically generated

Gambar 4. 19 Grafik *coal* *flow* pada *coal feeder unit* 1A

Tabel 4. 7 Hasil pengujian *coal* *flow* pada *coal feeder*  unit 1A per 10 menit

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Tanggal** | **Waktu** | **Nilai**  ***Dashboard***  **(t/h)** | **Nilai**  **DCS**  **(m2)** | **Selisih** | ***Error* (%)** |
| 1. | 01/02/2024 | 00.00. | 8.58 | 8.58 | 0.00 | 0.00 |
| 2. | 01/02/2024 | 00.10 | 8.07 | 8.07 | 0.00 | 0.00 |
| 3. | 01/02/2024 | 00.20 | 8.24 | 8.24 | 0.00 | 0.00 |
| 4. | 01/02/2024 | 00.30 | 8.92 | 8.92 | 0.00 | 0.00 |
| 5. | 01/02/2024 | 00.40 | 7.93 | 7.93 | 0.00 | 0.00 |
| 6. | 01/02/2024 | 00.50 | 8.44 | 8.44 | 0.00 | 0.00 |
| 7. | 01/02/2024 | 01.00 | 8.82 | 8.82 | 0.00 | 0.00 |
| 8. | 01/02/2024 | 01.10 | 8.49 | 8.49 | 0.00 | 0.00 |
| 9. | 01/02/2024 | 01.20 | 8.59 | 8.59 | 0.00 | 0.00 |
| 10. | 01/02/2024 | 01.30 | 7.84 | 7.84 | 0.00 | 0.00 |
|  |  |  |  |  |  | 0 |

Tabel diatas menampilkan hasil pengujian jumlah *coal* *flow* pada *coal feeder*  unit 1 yang A yang membandingan 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%.

Tabel 4. 8 Hasil pengujian *coal* *flow* pada *coal feeder*  unit 1A per 30 menit

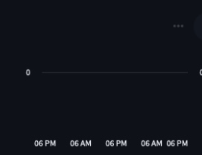
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Tanggal** | **Waktu** | **Nilai**  ***Dashboard***  **(t/h)** | **Nilai**  **DCS**  **(m2)** | **Selisih** | ***Error* (%)** |
| 1. | 02/02/2024 | 12.00 | 9.16 | 9.16 | 0.03 | 0.04 |
| 2. | 02/02/2024 | 12.30 | 8.29 | 8.29 | 0.02 | 0.02 |
| 3. | 02/02/2024 | 13.00 | 8.84 | 8.84 | 0.02 | 0.02 |
| 4. | 02/02/2024 | 13.30 | 8.66 | 8.66 | 0.04 | 0.05 |
| 5. | 02/02/2024 | 14.00 | 8.78 | 8.78 | 0.00 | 0.00 |
| 6. | 02/02/2024 | 14.30 | 9.18 | 9.18 | 0.00 | 0.00 |
| 7. | 02/02/2024 | 15.00 | 9.4 | 9.42 | 0.02 | 0.02 |
| 8. | 02/02/2024 | 15.30 | 9.41 | 9.41 | 0.00 | 0.00 |
| 9. | 02/02/2024 | 16.00 | 8.67 | 8.67 | 0.00 | 0.00 |
| 10. | 02/02/2024 | 16.30 | 9.62 | 9.62 | 0.02 | 0.02 |
|  |  |  |  |  |  | 0,1 |

Tabel diatas menampilkan hasil pengujian jumlah *coal* *flow* pada *coal feeder*  unit 1 yang A yang membandingan nilai dari *dashboard monitoring* dengan nilai dari DCS (*Distributed Control System*). Dari tabel diatas dapat disimpulkan bahwa dalam 10 kali percobaan dalam 30 menit rata-rata persentase *error* sistem *dashboard monitoring* sebesar 0.1%.

Tabel 4. 9 Hasil pengujian *coal* *flow* pada *coal feeder*  unit 1A per 1 jam

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Tanggal** | **Waktu** | **Nilai**  ***Dashboard***  **(t/h)** | **Nilai**  **DCS**  **(m2)** | **Selisih** | ***Error* (%)** |
| 1. | 03/02/2024 | 14.00 | 11.95 | 11.95 | 0.00 | 0.00 |
| 2. | 03/02/2024 | 15.00 | 11.92 | 11.92 | 0.00 | 0.00 |
| 3. | 03/02/2024 | 16.00 | 12.69 | 12.69 | 0.00 | 0.00 |
| 4. | 03/02/2024 | 17.00 | 11.11 | 11.11 | 0.00 | 0.00 |
| 5. | 03/02/2024 | 18.00 | 10.73 | 10.73 | 0.00 | 0.00 |
| 6. | 03/02/2024 | 19.00 | 11.99 | 11.99 | 0.00 | 0.00 |
| 7. | 03/02/2024 | 20.00 | 12.99 | 12.99 | 0.00 | 0.00 |
| 8. | 03/02/2024 | 21.00 | 12.82 | 12.82 | 0.00 | 0.00 |
| 9. | 03/02/2024 | 22.00 | 15.17 | 15.17 | 0.00 | 0.00 |
| 10. | 03/02/2024 | 23.00 | 15.23 | 15.23 | 0.00 | 0.00 |
|  |  |  |  |  |  | 0 |

Tabel diatas menampilkan hasil pengujian jumlah *coal* *flow* pada *coal feeder*  unit 1 yang A yang membandingan nilai dari *dashboard monitoring* dengan nilai dari DCS (*Distributed Control System*). Dari tabel diatas dapat disimpulkan bahwa dalam 10 kali percobaan dalam 1 jam rata-rata persentase *error* sistem *dashboard monitoring* sebesar 0%.



Gambar 4. 20 Grafik *coal* *flow* pada *coal feeder unit* 2A

Tabel 4. 10 Hasil pengujian *coal* *flow* pada *coal feeder*  unit 2A per 10 menit

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Tanggal** | **Waktu** | **Nilai**  ***Dashboard***  **(t/h)** | **Nilai**  **DCS**  **(m2)** | **Selisih** | ***Error* (%)** |
| 1. | 01/02/2024 | 00.00. | 0.00 | 0.00 | 0.00 | 0.00 |
| 2. | 01/02/2024 | 00.10 | 0.00 | 0.00 | 0.00 | 0.00 |
| 3. | 01/02/2024 | 00.20 | 0.00 | 0.00 | 0.00 | 0.00 |
| 4. | 01/02/2024 | 00.30 | 0.00 | 0.00 | 0.00 | 0.00 |
| 5. | 01/02/2024 | 00.40 | 0.00 | 0.00 | 0.00 | 0.00 |
| 6. | 01/02/2024 | 00.50 | 0.00 | 0.00 | 0.00 | 0.00 |
| 7. | 01/02/2024 | 01.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 8. | 01/02/2024 | 01.10 | 0.00 | 0.00 | 0.00 | 0.00 |
| 9. | 01/02/2024 | 01.20 | 0.00 | 0.00 | 0.00 | 0.00 |
| 10. | 01/02/2024 | 01.30 | 0.00 | 0.00 | 0.00 | 0.00 |
|  |  |  |  |  |  | 0 |

Tabel diatas menampilkan hasil pengujian jumlah *coal* *flow* pada *coal feeder*  unit 2 yang A yang membandingan 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%.

Tabel 4. 11 Hasil pengujian *coal* *flow* pada *coal feeder*  unit 2A per 30 menit

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Tanggal** | **Waktu** | **Nilai**  ***Dashboard***  **(t/h)** | **Nilai**  **DCS**  **(m2)** | **Selisih** | ***Error* (%)** |
| 1. | 02/02/2024 | 12.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 2. | 02/02/2024 | 12.30 | 0.00 | 0.00 | 0.00 | 0.00 |
| 3. | 02/02/2024 | 13.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 4. | 02/02/2024 | 13.30 | 0.00 | 0.01 | 0.01 | 0.01 |
| 5. | 02/02/2024 | 14.00 | 0.00 | 0.01 | 0.01 | 0.01 |
| 6. | 02/02/2024 | 14.30 | 0.00 | 0.01 | 0.01 | 0.01 |
| 7. | 02/02/2024 | 15.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 8. | 02/02/2024 | 15.30 | 0.00 | 0.00 | 0.00 | 0.00 |
| 9. | 02/02/2024 | 16.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 10. | 02/02/2024 | 16.30 | 0.00 | 0.00 | 0.00 | 0.00 |
|  |  |  |  |  |  | 0,1 |

Tabel diatas menampilkan hasil pengujian jumlah *coal* *flow* pada *coal feeder*  unit 2 yang A yang membandingan nilai dari *dashboard monitoring* dengan nilai dari DCS (*Distributed Control System*). Dari tabel diatas dapat disimpulkan bahwa dalam 10 kali percobaan dalam 30 menit rata-rata persentase *error* sistem *dashboard monitoring* sebesar 0.1%.

Tabel 4. 12 Hasil pengujian *coal* *flow* pada *coal feeder*  unit 2A per 1 jam

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Tanggal** | **Waktu** | **Nilai**  ***Dashboard***  **(t/h)** | **Nilai**  **DCS**  **(m2)** | **Selisih** | ***Error* (%)** |
| 1. | 03/02/2024 | 14.00 | 0.01 | 0.01 | 0.00 | 0.00 |
| 2. | 03/02/2024 | 15.00 | 0.01 | 0.01 | 0.00 | 0.00 |
| 3. | 03/02/2024 | 16.00 | 0.01 | 0.01 | 0.00 | 0.00 |
| 4. | 03/02/2024 | 17.00 | 0.01 | 0.01 | 0.00 | 0.00 |
| 5. | 03/02/2024 | 18.00 | 0.01 | 0.01 | 0.00 | 0.00 |
| 6. | 03/02/2024 | 19.00 | 0.01 | 0.01 | 0.00 | 0.00 |
| 7. | 03/02/2024 | 20.00 | 0.01 | 0.01 | 0.00 | 0.00 |
| 8. | 03/02/2024 | 21.00 | 0.01 | 0.01 | 0.00 | 0.00 |
| 9. | 03/02/2024 | 22.00 | 0.01 | 0.01 | 0.00 | 0.00 |
| 10. | 03/02/2024 | 23.00 | 0.01 | 0.01 | 0.00 | 0.00 |
|  |  |  |  |  |  | 0 |

Tabel diatas menampilkan hasil pengujian jumlah *coal* *flow* pada *coal feeder*  unit 2 yang A yang membandingan nilai dari *dashboard monitoring* dengan nilai dari DCS (*Distributed Control System*). Dari tabel diatas dapat disimpulkan bahwa dalam 10 kali percobaan dalam 1 jam rata-rata persentase *error* sistem *dashboard monitoring* sebesar 0%.

# 

# BAB V PENUTUP

## 5.1 Kesimpulan

Kesimpulan yang dapat diambil dari penelitian ini adalah perancangan sistem *monitoring* ini dapat berjalan dengan baik dan penambahan nilai parameter baru berhasil diimplementasikan dengan menampilkan stastus perlatalan dan nilai nilai parameter yang ada di *coal handling system* dengan rata-rata presentase *error* sebesar 0.1% dan memilik tingkat akurasi sebesar 99.9%. dengan tampilan *web* yang dapat diakses oleh semua tingkatan sepeerti manajemen, supervisor maupun staf kantor lainnya.

## 5.2 Saran

Berdasarkan kesimpulan diatas, dapat disarankan sebagai berikut ini:

1. Menambahkan fitur cetak dan *log* untuk memberikan pengguna akses untuk mencetak laporan dan memantau aktivitas sistem.
2. Implementasikan integrasi algoritma prediksi dan tampilkan hasil prediksi dengan tampilan visual yang mudah dimengerti untuk mendukung pengambilan keputusan yang tepat.
3. Memastikan fitur prediksi diuji cermat dan adopsi mekanisme notifikasi untuk memberi peringatan jika terdeteksi potensi masalah dalam sistem *monitoring* PLTU.

# DAFTAR PUSTAKA

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

[2] S. Alas Majapahit, A. Somantri, R. Ilham Subagja, U. Pasundan Bandung, and S. Mardira Indonesia, “Model Dashboard Monitoring Kinerja Pegawai Pada Top Level Management,” *J. Comput. Bisnis*, vol. 15, no. 2, pp. 108–114, 2021.

[3] 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.

[4] 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

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] A. Herliana and P. M. Rasyid, “Sistem Informasi Monitoring Pengembangan Software Pada Tahap,” *J. Inform.*, no. 1, pp. 41–50, 2016.

[10] I. D. Pranowo, Y. B. T. Bagastama, and T. A. F. Wibisono, “Communication between PLC different vendors using OPC server improved with application device,” *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 18, no. 3, pp. 1491–1498, 2020, doi: 10.12928/TELKOMNIKA.v18i3.14757.

[11] M. Nicola, C.-I. Nicola, M. Duță, and D. Sacerdoțianu, “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.

[12] 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.

[13] 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.

[14] 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).

[15] 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.

[16] F. NKD, “Memahami Pengertian Python dan Mengetahui Kegunaannya,” *LOGIQUE*, 2020.

[17] 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

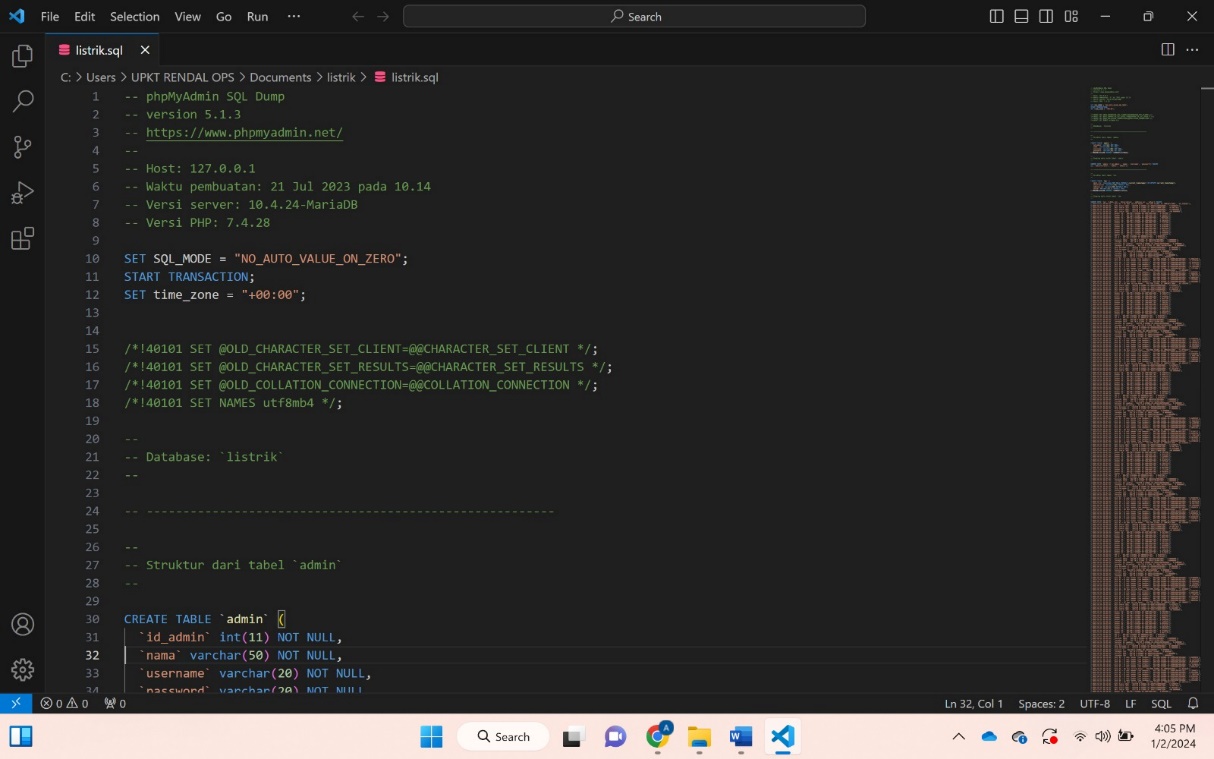
[18] M. Ariffudin, “Mengenal Visual Code Studio dan Fitur-Fitur Pentingnya,” *NIAGAHOSTERblog*, 2022. https://www.niagahoster.co.id/blog/visual-code-studio/ (accessed May 21, 2023).

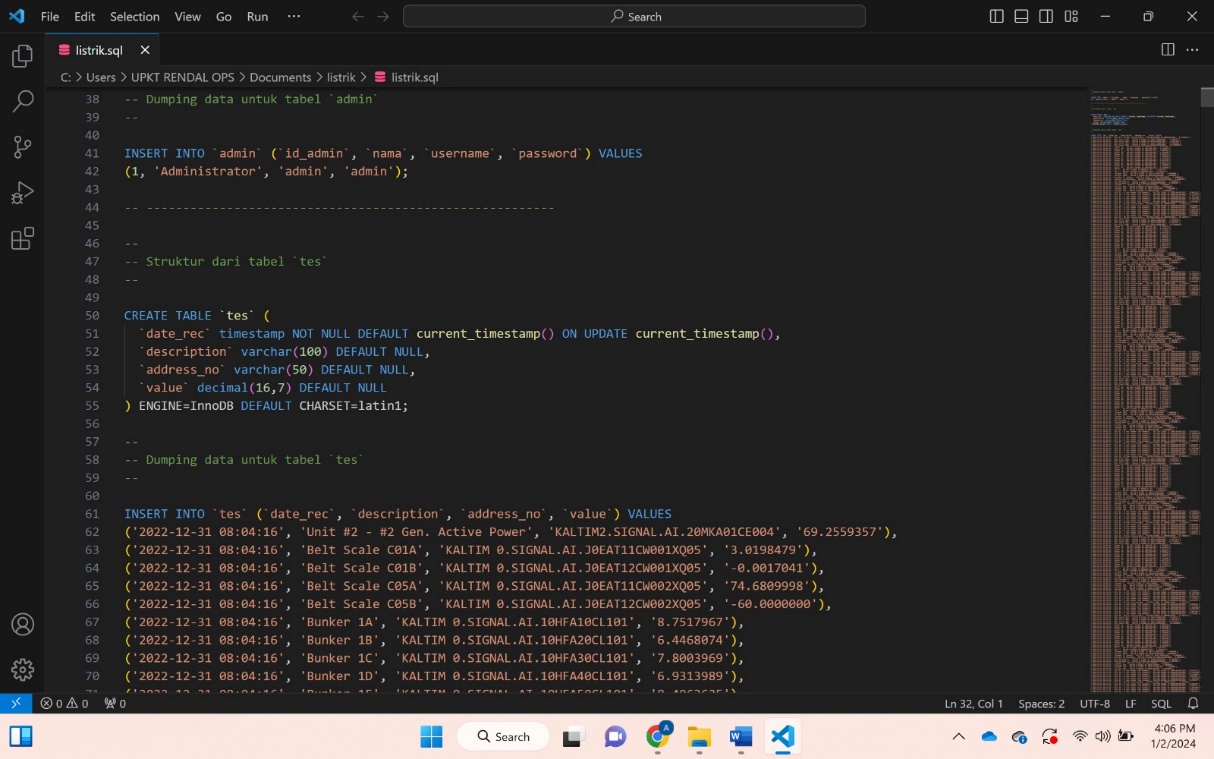
[19] F. Susanto, N. K. Prasiani, and P. Darmawan, “Implementasi Internet of Things Dalam Kehidupan Sehari-Hari,” *J. Imagine*, vol. 2, no. 1, pp. 35–40, 2022, doi: 10.35886/imagine.v2i1.329.

[20] T. Terkini, “Penjelasan dan Cara Kerja Konsep Internet of Things,” *mobnasesemka*, 2016. https://mobnasesemka.com/internet-of-things/ (accessed May 05, 2023).

# LAMPIRAN

1. Source data *coal handling system* pada PLTU Kaltim Teluk





1. Source code

|  |
| --- |
| #Modul Python  import pandas as pd                 # digunakan untuk pengolahan data frames, data tabels, data array. data pivot  import psycopg2                     #ini modul database opc  from psycopg2 import Error  from datetime import datetime, timedelta  import numpy as np  import streamlit as st  #Koneksi Modul  try:      connection =      print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")      print("OPC UJKT ................... Koneksi Tersambung")      print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")  except (Exception, Error) as error:      print("Error cek koneksi dan pastikan satu Jaringan Kantor", error)  #List Taglist DCS  lst=[]  MW1 = "'KALTIM1.SIGNAL.AI.10MKA01CE004'"  MW2 = "'KALTIM2.SIGNAL.AI.20MKA01CE004'"  SUA = "'KALTIM 0.SIGNAL.DI.J0EAA01AF001XB54'"  SUB = "'KALTIM 0.SIGNAL.DI.J0EAA02AF001XB54'"  C1A = "'KALTIM 0.SIGNAL.DI.J0EAC11AF001XB01'"  C1B = "'KALTIM 0.SIGNAL.DI.J0EAC12AF001XB01'"  C2U = "'KALTIM 0.SIGNAL.DI.J0EAC10AF002XB00'"  C2L = "'KALTIM 0.SIGNAL.DI.J0EAC10AF002XB01'"  C3 = "'KALTIM 0.SIGNAL.DI.J0EAC10AF003'"  C4A = "'KALTIM 0.SIGNAL.DI.J0EAC11AF004'"  C4B = "'KALTIM 0.SIGNAL.DI.J0EAC12AF004'"  CRA = "'KALTIM 0.SIGNAL.DI.J0EBC11AJ001XB00'"  CRB = "'KALTIM 0.SIGNAL.DI.J0EBC12AJ001XB00'"  C5A = "'KALTIM 0.SIGNAL.DI.J0EAC11AF005XB01'"  C5B = "'KALTIM 0.SIGNAL.DI.J0EAC12AF005XB01'"  C6A = "'KALTIM 0.SIGNAL.DI.J0EAC11AF006'"  C6B = "'KALTIM 0.SIGNAL.DI.J0EAC12AF006'"  B1A = 'KALTIM\_0.SIGNAL.AI.10HFA10CL101'  B1B = 'KALTIM\_0.SIGNAL.AI.10HFA20CL101'  B1C = 'KALTIM\_0.SIGNAL.AI.10HFA30CL101'  B1D = 'KALTIM\_0.SIGNAL.AI.10HFA40CL101'  B1E = 'KALTIM\_0.SIGNAL.AI.10HFA50CL101'  B1F = 'KALTIM\_0.SIGNAL.AI.10HFA60CL101'  B2A = 'KALTIM\_0.SIGNAL.AI.20HFA10CL101'  B2B = 'KALTIM\_0.SIGNAL.AI.20HFA20CL101'  B2C = 'KALTIM\_0.SIGNAL.AI.20HFA30CL101'  B2D = 'KALTIM\_0.SIGNAL.AI.20HFA40CL101'  B2E = 'KALTIM\_0.SIGNAL.AI.20HFA50CL101'  B2F = 'KALTIM\_0.SIGNAL.AI.20HFA60CL101'  CF1A = 'KALTIM1.SIGNAL.AI.10HFB10AF001XQ02'  CF1B = 'KALTIM1.SIGNAL.AI.10HFB20AF001XQ02'  CF1C = 'KALTIM1.SIGNAL.AI.10HFB30AF001XQ02'  CF1D = 'KALTIM1.SIGNAL.AI.10HFB40AF001XQ02'  CF1E = 'KALTIM1.SIGNAL.AI.10HFB50AF001XQ02'  CF1F = 'KALTIM1.SIGNAL.AI.10HFB60AF001XQ02'  CF2A = 'KALTIM2.SIGNAL.AI.20HFB10AF001XQ02'  CF2B = 'KALTIM2.SIGNAL.AI.20HFB20AF001XQ02'  CF2C = 'KALTIM2.SIGNAL.AI.20HFB30AF001XQ02'  CF2D = 'KALTIM2.SIGNAL.AI.20HFB40AF001XQ02'  CF2E = 'KALTIM2.SIGNAL.AI.20HFB50AF001XQ02'  CF2F = 'KALTIM2.SIGNAL.AI.20HFB60AF001XQ02'  lstCF = []  listCF = [CF1A, CF1B, CF1C, CF1D, CF1E, CF1F, CF2A, CF2B, CF2C, CF2D, CF2E, CF2F]  st.sidebar.header('Input Tanggal')  tglstart = st.sidebar.date\_input("Masukkan tanggal mulai")  tmstart = st.sidebar.time\_input("Masukkan waktu mulai")  tglover = st.sidebar.date\_input("Masukkan tanggal berakhir")  tmover = st.sidebar.time\_input("Masukkan waktu berakhir")  start = "'" + str(tglstart) + " " + str(tmstart) + "'"  finish = "'" + str(tglover) + " " + str(tmover) + "'"  #Data yang akan dipanggil  lst=[]  list = [MW1, MW2, SUA, SUB, C1A, C1B, C2U, C2L, C3, C4A, C4B, CRA, CRB, C5A, C5B, C6A, C6B]  bnk=[]  bunker = [B1A, B1B, B1C,B1D, B1E, B1F, B2A, B2B, B2C, B2D, B2E, B2F]  tglin = start  tglout = finish  menit="0,5,10,15,20,25,30,35,40,45,50,55"  curr\_now=datetime.now()  saiki =curr\_now.strftime("%d %B %Y")  col1, col2 = st.columns([3,1])  moving\_text = f"Skripsi iOT - {saiki}"  col1.write(      f"""      <div style="color: red; font-size: 24px;" id="moving-text">          {moving\_text}      </div>      <style>      @keyframes moveText {{          0% {{ transform: translateX(100%); }}          100% {{ transform: translateX(-100%); }}      }}      #moving-text {{          animation: moveText 10s linear infinite;      }}      </style>      """,      unsafe\_allow\_html=True,  )  col2.image('uniba.png', width=100)  #Resume  print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")  print("Input Tag DCS berjumlah",len(list),"Item Tag")  print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")  #query  print("Tarikan OPC Loading.....")  # =================================== DATA REALTIME ========================================  sql= connection.cursor()  query="select b.description, round(a.value,2) as nilai, b.satuan from current a,address b "  func="where a.address\_no="  dat=" and a.address\_no =b.address\_no"  for lop in list:      var = query +func +lop + dat      sql.execute(var)      lst.extend(sql.fetchall())  df = pd.DataFrame.from\_records(lst, columns=[x[0] for x in sql.description])  #Pivot of DataFrame Query SQL  data=df.pivot(index=None,columns='description',values='nilai').astype(float)  # =================================== DATA BUNKER ==========================================  sql2= connection.cursor()  query2="select a.date\_rec as tanggal\_penarikan,b.address\_no,b.description, round(a.value,2) as nilai\_operasi,b.satuan from history a,address b where a.address\_no="  func2=" and a.address\_no=b.address\_no "  dat2="and a.date\_rec >= timestamp "+tglin+" and a.date\_rec <= timestamp "+tglout+" "  ext2="and extract (minute from a.date\_rec) in ("+menit+") order by a.date\_rec"  #tag disesuaikan dengan taglis, silahkan tambahkan sesuaikan dengan keinginan  #loop Query  if menit=="0":      for lop2 in bunker:          var2=query2 + "'" + lop2 + "'" + func2 + dat2 + ext2          sql2.execute(var2)          bnk.extend(sql2.fetchall())  else:      for lop2 in bunker:          var2=query2 + "'" + lop2 + "'" + func2 + dat2 + ext2          sql2.execute(var2.replace('KALTIM\_0','KALTIM 0'))          bnk.extend(sql2.fetchall())  df2 = pd.DataFrame.from\_records(bnk, columns=[x[0] for x in sql2.description])  # =================================== DATA COAL FEEDER ==========================================  sql3= connection.cursor()  query3="select a.date\_rec as tanggal\_penarikan,b.address\_no,b.description, round(a.value,2) as nilai\_operasi,b.satuan from history a,address b where a.address\_no="  func3=" and a.address\_no=b.address\_no "  dat3="and a.date\_rec >= timestamp "+tglin+" and a.date\_rec <= timestamp "+tglout+" "  ext3="and extract (minute from a.date\_rec) in ("+menit+") order by a.date\_rec"  #tag disesuaikan dengan taglis, silahkan tambahkan sesuaikan dengan keinginan  #loop Query  for lop3 in listCF:      var3=query3 + "'" + lop3 + "'" + func3 + dat3 + ext3      sql3.execute(var3)      lstCF.extend(sql3.fetchall())    cok = ['Coal Feeder 1A', 'Coal Feeder 1B', 'Coal Feeder 1C', 'Coal Feeder 1D', 'Coal Feeder 1E', 'Coal Feeder 1F']  df3 = pd.DataFrame.from\_records(lstCF, columns=[x[0] for x in sql3.description])  data = data.fillna(0)  MWA = data['Unit #1 - #1 Gen. Active Power'].sum()  MWB = data['Unit #2 - #2 Gen. Active Power'].sum()  tot\_SUA = data['Ship Unloader A'].sum()  tot\_SUB = data['Ship Unloader B'].sum()  tot\_C1A = data['Conveyor C01A'].sum()  tot\_C1B = data['Conveyor C01B'].sum()  tot\_C2L = data['Conveyor C2 Loading'].sum()  tot\_C2U = data['Conveyor C2 Unloading'].sum()  tot\_C3 = data['Conveyor 3'].sum()  tot\_C4A = data['Conveyor C4A'].sum()  tot\_C5A = data['Conveyor C5A'].sum()  tot\_C6A = data['Conveyor C6A'].sum()  # Membuat DataFrame dari Array NumPy  daw = np.array([[tot\_SUA, tot\_SUB, tot\_C1A, tot\_C1B, tot\_C2U, tot\_C2L, tot\_C3, tot\_C4A, tot\_C5A, tot\_C6A],])  dw = pd.DataFrame(daw, columns=['Ship Unloader A', 'Ship Unloader B', 'Conveyor C01A', 'Conveyor C01B', 'Conveyor C2 Unloading', 'Conveyor C2 Loading', 'Conveyor 3', 'Conveyor C4A', 'Conveyor C5A', 'Conveyor C6A'])#.astype(str)  dataaa=df2.pivot(index='tanggal\_penarikan',columns='description',values='nilai\_operasi').astype(float)  dataCF=df3.pivot(index='tanggal\_penarikan',columns='description',values='nilai\_operasi').astype(float)  st.title("""  🏂 Skripsi UNIBA    Anshori  197030717  """)    col1, col2, col3 = st.columns([1, 1, 1])  col1.text("MW Unit 1")  col1.markdown(MWA)  col2.text("MW Unit 2")  col2.markdown(MWB)  col3.text("Ship Unloader A")  if float(dw['Ship Unloader A']) == 1:      col3.markdown(          f'<div style="background-color:{"#DB343A"}; padding:5px; border-radius:3px;">' 'SU A RUN' '</div>', unsafe\_allow\_html=True)  else:      col3.markdown(          f'<div style="background-color:{"#58EA0F"}; padding:5px; border-radius:3px;">' 'SU A STOP' '</div>', unsafe\_allow\_html=True)  col1.text("Ship Unloader B")  if float(dw['Ship Unloader B']) == 1:      col1.markdown(          f'<div style="background-color:{"#DB343A"}; padding:8px; border-radius:3px;">' 'SU B RUN' '</div>', unsafe\_allow\_html=True)  else:      col1.markdown(          f'<div style="background-color:{"#58EA0F"}; padding:8px; border-radius:3px;">' 'SU B STOP' '</div>', unsafe\_allow\_html=True)  col2.text("Conveyor C01A")  if float(dw['Conveyor C01A']) == 1:      col2.markdown(          f'<div style="background-color:{"#DB343A"}; padding:8px; border-radius:3px;">' 'C01A RUN' '</div>', unsafe\_allow\_html=True)  else:      col2.markdown(          f'<div style="background-color:{"#58EA0F"}; padding:8px; border-radius:3px;">' 'C01A STOP' '</div>', unsafe\_allow\_html=True)  col3.text("Conveyor C01B")  if float(dw['Conveyor C01B']) == 1:      col3.markdown(          f'<div style="background-color:{"#DB343A"}; padding:8px; border-radius:3px;">' 'C01B RUN' '</div>', unsafe\_allow\_html=True)  else:      col3.markdown(          f'<div style="background-color:{"#58EA0F"}; padding:8px; border-radius:3px;">' 'C01B STOP' '</div>', unsafe\_allow\_html=True)  col1.text("Conveyor C2 Unloading")  if float(dw['Conveyor C2 Unloading']) == 1:      col1.markdown(          f'<div style="background-color:{"#DB343A"}; padding:8px; border-radius:3px;">' 'C2 UNLOADING RUN' '</div>', unsafe\_allow\_html=True)  else:      col1.markdown(          f'<div style="background-color:{"#58EA0F"}; padding:8px; border-radius:3px;">' 'C2 UNLOADING STOP' '</div>', unsafe\_allow\_html=True)  col2.text("Conveyor C2 Loading")  if float(dw['Conveyor C2 Loading']) == 1:      col2.markdown(          f'<div style="background-color:{"#DB343A"}; padding:8px; border-radius:3px;">' 'C2 LOADING RUN' '</div>', unsafe\_allow\_html=True)  else:      col2.markdown(          f'<div style="background-color:{"#58EA0F"}; padding:8px; border-radius:3px;">' 'C2 LOADING STOP' '</div>', unsafe\_allow\_html=True)  col3.text("Conveyor C3")  if float(dw['Conveyor 3']) == 1:      col3.markdown(          f'<div style="background-color:{"#DB343A"}; padding:8px; border-radius:3px;">' 'C3 RUN' '</div>', unsafe\_allow\_html=True)  else:      col3.markdown(          f'<div style="background-color:{"#58EA0F"}; padding:8px; border-radius:3px;">' 'C3 STOP' '</div>', unsafe\_allow\_html=True)  col1.text("Conveyor C4")  if float(dw['Conveyor C4A']) == 1:      col1.markdown(          f'<div style="background-color:{"#DB343A"}; padding:8px; border-radius:3px;">' 'C4 RUN' '</div>', unsafe\_allow\_html=True)  else:      col1.markdown(          f'<div style="background-color:{"#58EA0F"}; padding:8px; border-radius:3px;">' 'C4 STOP' '</div>', unsafe\_allow\_html=True)  col2.text("Conveyor C5")  if float(dw['Conveyor C5A']) == 1:      col2.markdown(          f'<div style="background-color:{"#DB343A"}; padding:8px; border-radius:3px;">' 'C5 RUN' '</div>', unsafe\_allow\_html=True)  else:      col2.markdown(          f'<div style="background-color:{"#58EA0F"}; padding:8px; border-radius:3px;">' 'C5 STOP' '</div>', unsafe\_allow\_html=True)  col3.text("Conveyor C6")  if float(dw['Conveyor C6A']) == 1:      col3.markdown(          f'<div style="background-color:{"#DB343A"}; padding:8px; border-radius:3px;">' 'C6 RUN' '</div>', unsafe\_allow\_html=True)  else:      col3.markdown(          f'<div style="background-color:{"#58EA0F"}; padding:8px; border-radius:3px;">' 'C6 STOP' '</div>', unsafe\_allow\_html=True)  st.write(dataaa)  st.write(dataCF)  col1, col2, col3 = st.columns([1, 1, 1])  col1.area\_chart(dataaa['Bunker 1A'], width = 100, height = 200)  col2.area\_chart(dataaa['Bunker 1B'], width = 100, height = 200)  col3.area\_chart(dataaa['Bunker 1C'], width = 100, height = 200)  col1.area\_chart(dataaa['Bunker 1D'], width = 100, height = 200)  col2.area\_chart(dataaa['Bunker 1E'], width = 100, height = 200)  col3.area\_chart(dataaa['Bunker 1F'], width = 100, height = 200)  col1.area\_chart(dataaa['Bunker 2A'], width = 100, height = 200)  col2.area\_chart(dataaa['Bunker 2B'], width = 100, height = 200)  col3.area\_chart(dataaa['Bunker 2C'], width = 100, height = 200)  col1.area\_chart(dataaa['Bunker 2D'], width = 100, height = 200)  col2.area\_chart(dataaa['Bunker 2E'], width = 100, height = 200)  col3.area\_chart(dataaa['Bunker 2F'], width = 100, height = 200)  col1.area\_chart(dataCF['Unit #1 - A coal feeder flow feedback'], width = 100, height = 200)  col2.area\_chart(dataCF['Unit #1 - B coal feeder flow feedback'], width = 100, height = 200)  col3.area\_chart(dataCF['Unit #1 - C coal feeder flow feedback'], width = 100, height = 200)  col1.area\_chart(dataCF['Unit #1 - D coal feeder flow feedback'], width = 100, height = 200)  col2.area\_chart(dataCF['Unit #1 - E coal feeder flow feedback'], width = 100, height = 200)  col3.area\_chart(dataCF['Unit #1 - F coal feeder flow feedback'], width = 100, height = 200)  col1.area\_chart(dataCF['Unit #2 - A coal feeder flow feedback'], width = 100, height = 200)  col2.area\_chart(dataCF['Unit #2 - B coal feeder flow feedback'], width = 100, height = 200)  col3.area\_chart(dataCF['Unit #2 - C coal feeder flow feedback'], width = 100, height = 200)  col1.area\_chart(dataCF['Unit #2 - D coal feeder flow feedback'], width = 100, height = 200)  col2.area\_chart(dataCF['Unit #2 - E coal feeder flow feedback'], width = 100, height = 200)  col3.area\_chart(dataCF['Unit #2 - F coal feeder flow feedback'], width = 100, height = 200) |