Perancangan Dan Implementasi *Dashboard* Iot Penggunaan Listrik Produksi Di Sdb Pt Astra Otoparts Berbasis Node-Red

1sr Friska Aulia Nurahman
Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia
friskeyyy@student.telkomuniversity.ac.id

2rd Denny Darlis
Fakultas Ilmu Terapan
Universitas Telkom
Bandung,Indonesia
denny.darlis@tass.telkomuniversity.ac.id

3rd Muhammad Iqbal
Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia
miqbal@tass.telkomuniversity.ac.id

***Abstrak* — Sistem *dashboard* berbasis *Internet of Things* (IoT) dirancang untuk memantau konsumsi energi listrik pada panel *Sub Distribution Board* (SDB) di PT Astra Otoparts. Node-RED digunakan sebagai platform akuisisi data, dengan protokol Modbus TCP/IP untuk komunikasi dengan kWh meter yang terpasang pada panel SDB 1, 2, 24, dan 25. Data yang diperoleh disimpan dalam database MySQL dan divisualisasikan melalui *dashboard* web menggunakan React.js. Arsitektur sistem mengikuti model *three-tier* yang terdiri dari lapisan presentasi, logika, dan data. Fitur pemantauan *real-time* dan historis memungkinkan pengguna untuk menganalisis pola konsumsi energi secara lebih akurat, sehingga mendukung efisiensi operasional dan pengambilan keputusan berbasis data. Hasil implementasi menunjukkan sistem mampu beroperasi secara stabil dalam kondisi *online* maupun *offline*, serta memberikan visualisasi data yang informatif. Integrasi IoT dalam sistem ini menunjukkan potensi besar dalam pengelolaan energi yang cerdas di lingkungan industri.**

Kata Kunci: Dashboard IoT, Node-RED, Modbus TCP/IP, kWh Meter, MySQL, React.js

I. PENDAHULUAN

Perkembangan teknologi *Internet of Things* (IoT) telah mendorong transformasi digital di berbagai sektor industri, termasuk dalam pengelolaan energi. IoT memungkinkan integrasi antara sensor, perangkat lunak, dan jaringan komunikasi untuk memperoleh data dari perangkat fisik secara *real-time*. Salah satu platform yang banyak digunakan untuk membangun sistem IoT adalah Node-RED, yang mendukung berbagai protokol komunikasi industri seperti Modbus TCP/IP serta menyediakan antarmuka visual yang memudahkan pengembangan sistem.

PT Astra Otoparts Divisi Nusametal merupakan perusahaan manufaktur komponen otomotif berbasis aluminium untuk kendaraan roda dua dan empat. Dalam mendukung efisiensi operasional, perusahaan memiliki Departemen Digitalization & Expert, khususnya *Section Digitalization Development* yang berfokus pada integrasi teknologi *Information Technology* (IT) dan *Operational Technology* (OT). Salah satu tantangan yang dihadapi adalah pemantauan konsumsi energi listrik pada panel *Sub Distribution Board* (SDB), yang hingga kini masih dilakukan secara manual.

Pemantauan manual menyebabkan keterlambatan dalam deteksi lonjakan beban, keterbatasan data historis, dan minimnya integrasi dengan sistem analisis lainnya. Kondisi ini berdampak pada potensi pemborosan energi dan peningkatan biaya operasional. Oleh karena itu, dibutuhkan sistem monitoring berbasis IoT yang mampu menampilkan data konsumsi energi secara real-time dan historis.

Penelitian ini bertujuan untuk merancang dan mengimplementasikan *Dashboard SDB* berbasis Node-RED guna memantau konsumsi energi listrik pada panel SDB di area produksi PT Astra Otoparts Divisi Nusametal. Sistem ini dilengkapi dengan visualisasi data berbentuk grafik, pembacaan kWh secara berkala, serta penyimpanan data historis untuk analisis tren konsumsi energi. Diharapkan solusi ini dapat meningkatkan efisiensi operasional, mengurangi pemborosan energi, dan meminimalisasi human error dalam proses pemantauan energi.

II. DASAR TEORI

A. Tinjauan Umum dan Sumber Dataset

Pada Implementasinya, Sistem pemantauan penggunaan energi listrik secara real-time, yang dapat memberikan gambaran langsung terhadap konsumsi daya pada berbagai titik produksi. Dalam tugas akhir ini, dirancang dan diimplementasikan sebuah dashboard IoT untuk memonitor penggunaan listrik produksi di Sub Distribution Board (SDB) PT Astra Otoparts, menggunakan platform Node-RED.

B. *Internet of Things* (IoT)

Internet of Things (IoT) adalah sebuah sistem teknologi yang memungkinkan perangkat fisik untuk terhubung satu sama lain dan bertukar informasi, memproses data secara mandiri, serta membuat keputusan secara otomatis tanpa perlu campur tangan langsung dari pengguna [3]. Teknologi ini bekerja dengan menggunakan sensor, perangkat lunak, dan jaringan komunikasi untuk mengumpulkan dan mengirimkan data secara *real-time* dari dunia nyata ke sistem digital.

C. *Modbus* TCP

Modbus TCP adalah protokol *Modbus* yang sistem pengirimannya tidak menggunakan komunikasi serial tetapi menggunakan pembungkus *TCP/IP* dan dikirimkan melalui arsitektur jaringan [4]. Protokol dalam penelitian ini diimplementasikan dalam jaringan *smart building* dengan menggunakan media transmisi kabel dan nirkabel. Protokol ini bekerja di atas *TCP/IP*, memungkinkan pertukaran data secara cepat dan stabil antara perangkat-perangkat tersebut, sehingga sangat cocok digunakan dalam sistem pemantauan jarak jauh dan *real-time*.

D. Node-RED

Node-RED merupakan aplikasi dengan *platform Node.js* yang berjalan di *server* dan melayani perangkat yang terhubung ke *server* tersebut. Namun, alat ini tidak hanya dapat mencakup *IoT* tetapi juga aplikasi web standar [5]. *Platform* ini menyediakan antarmuka visual yang memudahkan proses perancangan sistem otomasi tanpa perlu banyak menulis kode secara manual.

E. MySQL

MySQL adalah sistem manajemen basis data berbasis SQL *(Structured Query Language)* yang memungkinkan

pengguna untuk menyimpan, mengelola, dan mengambil data dengan cara yang terstruktur. Sebagai perangkat lunak *open-source*, MySQL memberikan kebebasan kepada pengembang untuk memodifikasi dan mendistribusikan perangkat lunak ini sesuai kebutuhan [6].

F. Perancangan *Backend* Aplikasi

React, atau sering disebut dengan *React.js* atau juga *ReactJS,* merupakan *JavaScript library* yang dikembangkan oleh *Facebook* untuk memfasilitasi daripada komponen pembuatan antarmuka dan interaktif serta mudah untuk digunakan ulang. *ReactJS* memiliki keunggulan dari segi kecepatan, *simplicity*, dan *sclability* [7]. *ReactJS* mendukung pengembangan *dashboard IoT* yang interaktif dan mudah digunakan, sehingga mempermudah pengguna dalam memantau dan menganalisis data konsumsi listrik secara efisien.

G. Node.JS

Node.js adalah sebuah *platform* yang memungkinkan untuk menjalankan kode JavaScript di sisi server, bukan hanya di *browser*. *Platform* ini dibangun menggunakan mesin V8 milik Google, yang membuat proses eksekusi kodenya cepat dan ringan. Salah satu kelebihan Node.js adalah kemampuannya menangani banyak permintaan secara bersamaan tanpa harus menunggu proses lain selesai lebih dulu, karena menggunakan sistem *event-driven.*

H. Visual Studio Code

Visual Studio Code adalah editor kode sumber yang dikembangkan oleh Microsoft. Perangkat lunak ini ringan, *cross-platform*, dan *open-source*. Microsoft pertama kali merilis *VS Code* pada April 2015. Dibangun di atas *Electron*, VS Code dirancang sebagai editor kode yang ringan dan bersifat *open-source*. Visual Studio Code sangat ideal untuk pengembangan aplikasi web modern, termasuk React.js, sebuah *library* JavaScript populer untuk membangun antarmuka pengguna (*User interface/UI)* yang dinamis dan *responsive*, serta membantu pengembang menulis kode *React* lebih efisien dan rapi.

III. METODE

A. Arsitektur Sistem

Dashboard *monitoring* konsumsi listrik pada *SDB* dirancang menggunakan arsitektur jaringan *client-server* dengan pendekatan *three-tier* [8]. Model ini membagi sistem menjadi tiga lapisan utama yaitu *presentation layer*, *logical layer*, dan *data layer*. Penggunaan model *three-tier* dipilih karena memberikan keunggulan dalam aspek keamanan, skalabilitas, serta mempermudah proses pemeliharaan dan pengembangan sistem secara modular.

B. Spesifikasi Perancangan

1. *Hardware*

Hardware atau perangkat keras adalah komponen fisik pada perangkat *computer* [9]. *Hardware* bertugas untuk menjalankan perintah yang telah ditentukan oleh pengguna [9]. Dalam lingkungan perusahaan, keberadaan perangkat keras sangat penting karena menjadi fondasi utama dalam menjalankan sistem informasi yang digunakan untuk mendukung operasional harian. Kinerja sistem sangat bergantung pada spesifikasi dan kondisi perangkat keras yang digunakan. Oleh karena itu, perusahaan perlu memastikan bahwa perangkat keras yang dimiliki memiliki performa yang memadai, stabil, dan dapat diandalkan untuk memproses data, menyimpan informasi, serta menjaga konektivitas antar unit kerja. Pada table 1 adalah kebutuhan hardware yang dibutuhkan.

TABEL 1
Kebutuhan Operasional Hardware

Hardware	**Spesifikasi**
Laptop/Komputer	Intel Core i5-1135G7
RAM	16 GB
penyimpanan	256 GB
Sistem Operasi	Windows 11
Kabel Data	Kabel LAN RJ45

TABEL 2
Kebutuhan Hardware (Lingkungan Operasional)

Hardware	**Spesifikasi**	**Gambar**
kWh meter	Mitsubishi *Electric* kWh Meter	
Panel *SDB*	*SDB* 1, 2, 24, dan 25 sebagai lokasi pemasangan perangkat pemantauan	
Gateway	Modbus TCP/IP *Ethernet*	
Jaringan Lokal	Infrastruktur jaringan lokal perusahaan	

2. *Software*

Software atau perangkat lunak adalah perangkat yang tidak berbentuk fisik, namun dapat dioperasikan oleh penggunanya [10]. Perangkat lunak berfungsi untuk menginput seluruh aktivitas komputer seperti menyimpan data, mengecek data, memanipulasi data, dan memperoleh dari data yang dilakukan oleh perangkat keras [10]. Perangkat lunak yang dipakai dalam proses pembuatan sistem *Monitoring SDB* dapat dilihat pada table dibawah ini:

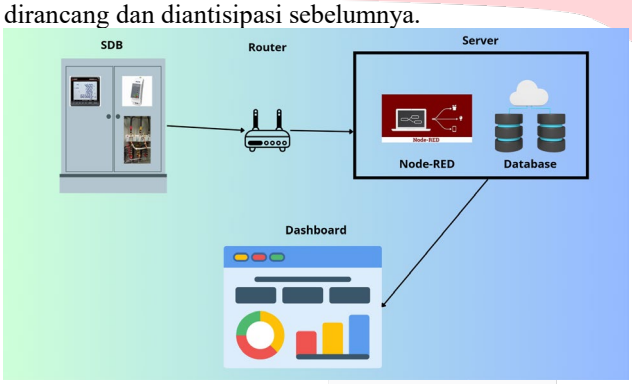
TABEL 3
Kebutuhan Operasional Software

Software	**Fungsi**

Node-RED	Akuisisi data Modbus TCP dan pengiriman ke MySQL
MySQL	Penyimpanan data historis konsumsi kWh
Node.js & Express.js	Pembuatan REST API sebagai jembatan data
React.js	Antarmuka pengguna
Visual Studio Code	IDE untuk pengembangan *frontend* dan *backend*
Google Chrome	*Web browser* untuk mengakses *dashboard*
DBeaver	GUI untuk manajemen *database* MySQL

C. Blok Diagram Sistem

Proses perancangan sistem aplikasi *mobile* presensi ini divisualisasikan melalui sebuah flowchart yang menggambarkan alur logika dan interaksi antara pengguna dengan sistem. Dengan adanya flowchart ini, pengembangan sistem menjadi lebih terstruktur karena setiap alur kerja telah dirancang dan diantisipasi sebelumnya.

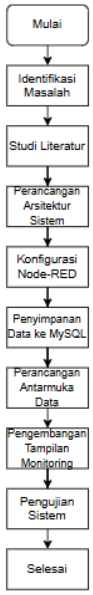


GAMBAR 1
Blok Diagram Sistem

Sistem ini dirancang untuk bekerja dalam jaringan local perusahaan. Dimulai dari *Sub Distribution Board (SDB)* yang dilengkapi *kWh meter* dan *IoT Gateway* yang berfungsi mencatat data konsumsi energi listrik secara langsung dari mesin produksi. Data tersebut dikirim menuju *laptop* yang menjalankan *platform Node-RED*. Di dalam *Node-RED*, alur pengambilan data disusun secara visual untuk membaca nilai *kWh* dari setiap *SDB* melalui protokol *Modbus TCP/IP*. Dilakukan secara bertahap dan terstruktur, dimulai dari tahap perencanaan awal, perancangan sistem, pengembangan fitur, pengujian aplikasi, hingga tahap akhir berupa publikasi. Setiap tahapan ditampilkan secara runtut untuk menunjukkan bagaimana proses berjalan secara sistematis. Pengembangan diawali dengan termin "Mulai" sebagai titik awal dari seluruh kegiatan, dan diakhiri pada langkah "Selesai" yang menandakan bahwa aplikasi telah melewati seluruh proses pengembangan, termasuk uji coba dan evaluasi, sehingga siap untuk digunakan oleh pengguna secara optimal.

D. Tahapan Perancangan

Proses perancangan Sistem *monitoring* kWh pada SDB *via* Wi-Fi menggunakan komunikasi modbus TCP/IP ini dapat dilihat pada gambar 3.2 dibawah ini.



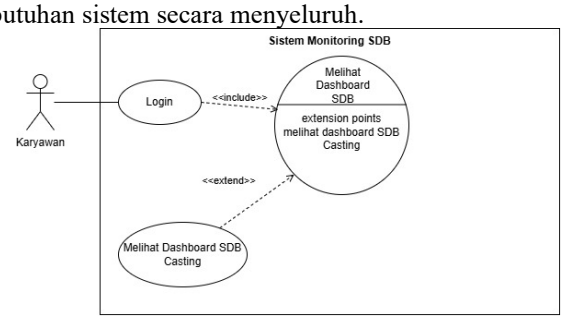
GAMBAR 2
Flowchart

Proses dimulai dengan menentukan kebutuhan sistem monitoring energi listrik pada panel SDB di PT Astra Otoparts. Permasalahan utama yang diidentifikasi adalah bahwa proses pemantauan konsumsi listrik di area produksi saat ini belum menggunakan sistem otomatis, sehingga informasi penggunaan daya tidak dapat diperoleh secara cepat dan akurat. Untuk mendukung pengembangan solusi, dilakukan studi literatur yang mengkaji teknologi relevan seperti Modbus TCP/IP, Node-RED, MySQL, serta sistem *dashboard* berbasis *web*. Selanjutnya, dilakukan perancangan arsitektur sistem yang mencakup blok diagram dan pemetaan data dari kWh meter menuju *database* dan tampilan antarmuka. Flow pada Node-RED disusun untuk membaca data dari kWh meter, memprosesnya menjadi data kWh, dan mengirimkannya ke *database*. Data yang diterima dari Node-RED akan disimpan secara otomatis ke dalam *database* MySQL untuk tiap panel. *Backend* sistem dirancang menggunakan Node.js untuk menyediakan data dalam bentuk *endpoint* API, sementara tampilan monitoring dikembangkan menggunakan React.js guna menampilkan data dalam bentuk grafik dan pembacaan kWh yang diperbarui secara *real-time*. Setelah seluruh komponen sistem dikembangkan, dilakukan pengujian terhadap alur komunikasi mulai dari perangkat hingga tampilan *frontend*, serta validasi kesesuaian data dengan tampilan fisik. Sistem dinyatakan selesai dan siap digunakan setelah semua komponen terintegrasi dengan baik.

E. Analisis UML (*Unified Modeling Language)*

Analisis kebutuhan fungsional sistem dilakukan menggunakan *Unified Modeling Language* (UML), yaitu bahasa pemodelan standar untuk menggambarkan, membangun, dan mendokumentasikan sistem perangkat lunak. UML digunakan untuk memodelkan struktur dan perilaku sistem dari tahap analisis hingga desain. Dalam pengembangan sistem monitoring SDB, analisis kebutuhan fungsional divisualisasikan melalui *use case* diagram. Diagram ini menggambarkan interaksi antara sistem dan aktor (pengguna) secara visual, serta digunakan pada tahap awal pengembangan untuk memahami kebutuhan fungsional dan mempermudah komunikasi antara analis dan pengguna. Elemen penting dalam *use case* diagram meliputi *actor*, *use*

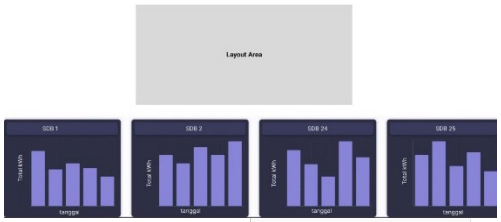
case, *system boundary*, serta hubungan *association*, *include*, dan *extend*. Dengan elemen-elemen tersebut, *use case* diagram menjadi alat utama dalam memahami interaksi dan kebutuhan sistem secara menyeluruh.



GAMBAR 3
Use Case Diagram

Gambar diatas merupakan *Use Case Diagram* dari sistem monitoring SDB yang memiliki aktor yaitu karyawan dengan akses dapat melihat *dashboard* SDB *casting*. Sebelum melihat SDB *casting*, karyawan harus login terlebih dahulu. *Login* ke *dashboard* hanya bisa dilakukan oleh staff, kepala seksi, manager, dan admin.

F. Tampilan Yang Diharapkan



GAMBAR 4
Mockup Dashboard

Tampilan antarmuka pengguna merupakan komponen penting dalam sistem pemantauan konsumsi listrik, dirancang agar informatif, mudah digunakan, dan responsif. Tujuan utamanya adalah mempermudah pengguna memahami data penggunaan energi dari tiap *Sub Distribution Board* (SDB). *Mockup dashboard* menampilkan *Layout* Area di tengah layar sebagai *overview* kondisi sistem produksi, termasuk mesin, peta lokasi SDB, dan jalur produksi. Di bagian bawah, terdapat grafik batang yang menunjukkan total konsumsi listrik harian (kWh) dari beberapa panel seperti SDB 1, 2, 24, dan 25, guna memudahkan analisis tren pemakaian listrik.

IV. HASIL DAN PEMBAHASAN

A. Hasil Perancangan

Sistem ini dirancang untuk memantau data konsumsi listrik secara *real-time*, menyimpannya secara historis, dan menyajikannya dalam bentuk visualisasi yang informatif. Dengan adanya sistem ini, perusahaan dapat memperoleh gambaran menyeluruh mengenai pola penggunaan energi listrik di area produksi, sehingga dapat mendukung upaya penghematan energi dan pengambilan keputusan berbasis data.

Arsitektur sistem menggunakan pendekatan terdistribusi, yang terdiri dari beberapa komponen utama yang saling terhubung. Sensor *kWh meter* digunakan sebagai perangkat pengukuran energi yang terhubung ke jaringan melalui protokol komunikasi *Modbus TCP/IP*. Data dari perangkat *kWh meter* kemudian diakuisisi menggunakan *Node-RED*, diproses, dan disimpan ke dalam *database MySQL*. Selanjutnya, data tersebut disediakan melalui *API* backend dan divisualisasikan melalui dashboard berbasis web menggunakan *framework React.js*.

Komponen utama yang digunakan dalam sistem ini meliputi:

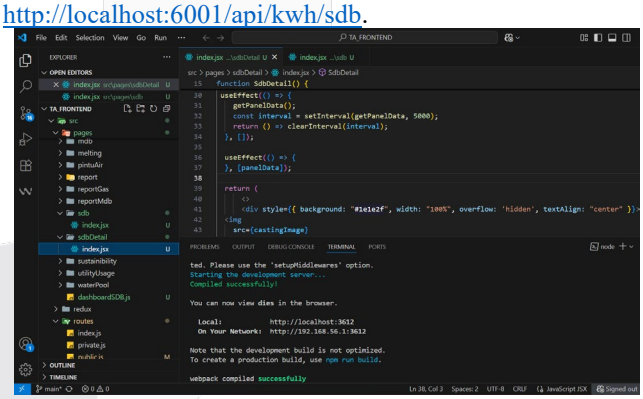
a. kWh meter sebagai alat pengukuran energi pada panel *SDB*
b. Modbus TCP/IP sebagai protokol komunikasi antara perangkat dan server.
c. Node-RED sebagai pengolahan dan alur data.
d. Node.js sebagai API penghubung antara *database* dan *frontend.*
e. React.js tampilan antarmuka pengguna berbasis *web.*
f. Laptop/PC sebagai pemograman serta pemantauan sistem.

Sistem ini difokuskan pada empat panel SDB yaitu SDB 1, SDB 2,SDB 24,dan SDB 25, menyesuaikan dengan ketersediaan perangkat kWh meter di area produksi. Setiap *flow* di Node-RED dirancang untuk membaca data dari perangkat, memprosesnya menjadi nilai kWh yang dapat ditampilkan, dan menyimpannya ke masing-masing tabel di *database*.

Struktur *Script* dibagi menjadi dua bagian, yaitu:

GAMBAR 5
Script Backend

backend sebagai tempat *backend API* dijalankan untuk menyajikan data melalui *endpoint* seperti http://localhost:6001/api/kwh/sdb.



GAMBAR 6
Script Frontend

frontend untuk menampilkan data dalam bentuk grafik menggunakan empat API yang sesuai dengan masing-masing SDB.

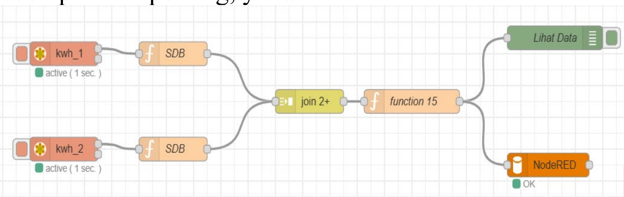
Tahap akhir dari perancangan dan implementasi sistem ini adalah memastikan bahwa seluruh komponen dapat berjalan dengan baik dan saling terintegrasi. Setelah semua *flow* di *deploy* di Node-RED, data dari kWh meter berhasil diambil dan tersimpan ke *database*.

Dashboard kemudian menampilkan grafik konsumsi energi dari tiap SDB secara *real-time*, serta menyediakan informasi total kWh untuk setiap panel. Sistem juga dirancang agar dapat berjalan secara *offline* melalui jaringan lokal perusahaan, sehingga tetap dapat digunakan tanpa koneksi internet.

B. Hasil Implementasi

Setelah proses perancangan selesai, sistem diuji coba secara langsung dengan menghubungkan laptop ke jaringan lokal menggunakan kabel LAN atau Wi-Fi. Koneksi ini diperlukan agar laptop dapat terhubung ke perangkat kWh meter yang sudah dipasang pada panel SDB 1, SDB 2, SDB 24, dan SDB 25 melalui protokol komunikasi Modbus TCP/IP.

Node-RED dijalankan melalui terminal, lalu diakses melalui *browser*. Di dalam Node-RED, dibuat empat *flow* terpisah untuk masing-masing panel SDB. Setiap *flow* terdiri dari beberapa *node* penting, yaitu:

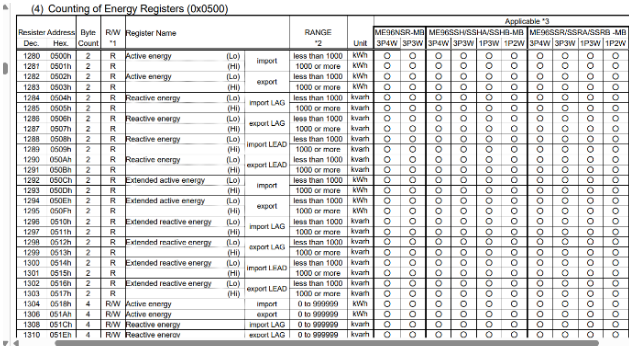


GAMBAR 7
Flow SDB 1

TABEL 4
Jenis node pada flow

Nama Node	Jenis Node	Fungsi
kwh_1	*Modbus Read*	Membaca data pecahan desimal dari *register* 1281 setiap 1 detik
Kwh_2	*Modbus Read*	Membaca data pecahan desimal dari *register* 1280 setiap 1 detik
SDB	*Function*	Menandai data berasal dari panel *SDB* tertentu
Join 2+	*Join*	Menggabungkan dua input menjadi satu data objek
Function 15	*Function*	Menggabungkan nilai *register* menjadi format desimal seperti 566.788
Lihat Data	*Debug/Display*	Menampilkan output data untuk monitoring
NodeRED	*MySQL*	Mengirim dan menyimpan data ke tabel *database MySQL*

Data *kWh* diambil menggunakan dua alamat *register*, yaitu 1281 dan 1280. Register 1281 menyimpan nilai bilangan bulat dari *kWh*, sedangkan register 1280 menyimpan bagian desimalnya. Nilai dari kedua *register* ini kemudian dikombinasikan untuk menghasilkan total energi dalam format desimal, contohnya: 568 dan 725 menjadi 568.725 *kWh*. Informasi tentang penggunaan register ini mengacu pada datasheet resmi power meter Mitsubishi, yang telah dijadikan dasar dalam konfigurasi sistem.



(4) Counting of Energy Registers (0x0500)

Resister Address		Byte	R/W	Register Name			RANGE		Applicable *3									
									ME96NSR-MB		ME96SSH/SSHA/SSHB-MB				ME96SSR/SSRA/SSRB -MB			
Dec.	Hex.	Count	*1				*2	Unit	3P4W	3P3W	3P4W	3P3W	1P3W	1P2W	3P4W	3P3W	1P3W	1P2W
1280	0500h	2	R	Active energy	(Lo)	import	less than 1000	kWh	○	○	○	○	○	○	○	○	○	○
1281	0501h	2	R		(Hi)		1000 or more	kWh	○	○	○	○	○	○	○	○	○	○
1282	0502h	2	R	Active energy	(Lo)	export	less than 1000	kWh	○	○	○	○	○	○	○	○	○	○
1283	0503h	2	R		(Hi)		1000 or more	kWh	○	○	○	○	○	○	○	○	○	○
1284	0504h	2	R	Reactive energy	(Lo)	import LAG	less than 1000	kvarh	○	○	○	○	○	○	○	○	○	○
1285	0505h	2	R		(Hi)		1000 or more	kvarh	○	○	○	○	○	○	○	○	○	○
1286	0506h	2	R	Reactive energy	(Lo)	export LAG	less than 1000	kvarh	○	○	○	○	○	○	○	○	○	○
1287	0507h	2	R		(Hi)		1000 or more	kvarh	○	○	○	○	○	○	○	○	○	○
1288	0508h	2	R	Reactive energy	(Lo)	import LEAD	less than 1000	kvarh	○	○	○	○	○	○	○	○	○	○
1289	0509h	2	R		(Hi)		1000 or more	kvarh	○	○	○	○	○	○	○	○	○	○
1290	050Ah	2	R	Reactive energy	(Lo)	export LEAD	less than 1000	kvarh	○	○	○	○	○	○	○	○	○	○
1291	050Bh	2	R		(Hi)		1000 or more	kvarh	○	○	○	○	○	○	○	○	○	○
1292	050Ch	2	R	Extended active energy	(Lo)	import	less than 1000	kWh	○	○	○	○	○	○	○	○	○	○
1293	050Dh	2	R		(Hi)		1000 or more	kWh	○	○	○	○	○	○	○	○	○	○
1294	050Eh	2	R	Extended active energy	(Lo)	export	less than 1000	kWh	○	○	○	○	○	○	○	○	○	○
1295	050Fh	2	R		(Hi)		1000 or more	kWh	○	○	○	○	○	○	○	○	○	○
1296	0510h	2	R	Extended reactive energy	(Lo)	import LAG	less than 1000	kvarh	○	○	○	○	○	○	○	○	○	○
1297	0511h	2	R		(Hi)		1000 or more	kvarh	○	○	○	○	○	○	○	○	○	○
1298	0512h	2	R	Extended reactive energy	(Lo)	export LAG	less than 1000	kvarh	○	○	○	○	○	○	○	○	○	○
1299	0513h	2	R		(Hi)		1000 or more	kvarh	○	○	○	○	○	○	○	○	○	○
1300	0514h	2	R	Extended reactive energy	(Lo)	import LEAD	less than 1000	kvarh	○	○	○	○	○	○	○	○	○	○
1301	0515h	2	R		(Hi)		1000 or more	kvarh	○	○	○	○	○	○	○	○	○	○
1302	0516h	2	R	Extended reactive energy	(Lo)	export LEAD	less than 1000	kvarh	○	○	○	○	○	○	○	○	○	○
1303	0517h	2	R		(Hi)		1000 or more	kvarh	○	○	○	○	○	○	○	○	○	○
1304	0518h	4	R/W	Active energy		import	0 to 999999	kWh	○	○	○	○	○	○	○	○	○	○
1306	051Ah	4	R/W	Active energy		export	0 to 999999	kWh	○	○	○	○	○	○	○	○	○	○
1308	051Ch	4	R/W	Reactive energy		import LAG	0 to 999999	kvarh	○	○	○	○	○	○	○	○	○	○
1310	051Eh	4	R/W	Reactive energy		export LAG	0 to 999999	kvarh	○	○	○	○	○	○	○	○	○	○

GAMBAR 8
datasheet power meter Mitsubishi

Konfigurasi koneksi untuk setiap perangkat kWh meter disesuaikan berdasarkan identitas masing-masing perangkat, meskipun beberapa perangkat terletak pada jaringan yang sama dan menggunakan alamat IP yang sama. Dengan pengaturan tersebut, sistem tetap mampu membedakan dan membaca data dari setiap panel SDB secara tepat dan jelas. Setelah *flow* berhasil di-*deploy*, sistem mulai menyimpan data ke dalam *database* MySQL secara teratur setiap detik. Hasil pengujian menunjukkan bahwa nilai yang dikirim oleh masing-masing panel tetap konsisten dan stabil, sesuai dengan data yang tampil pada layar perangkat kWh meter. Selain itu, hasil visualisasi di *dashboard* juga menampilkan grafik yang sesuai dengan pembacaan perangkat, sehingga menunjukkan alur komunikasi, proses pemrosesan, dan penyimpanan data berjalan dengan baik sesuai harapan.

GAMBAR 9
Debug/keluaran dari flow Node-RED

Selanjutnya, *script backend* dijalankan untuk menyediakan API menggunakan Node.js dan Express.js. API ini memungkinkan data dari database diakses melalui *endpoint* seperti http://localhost:6001/api/kwh/sdb1, http://localhost:6001/api/kwh/sdb2,http://localhost:6001/api/kwh/sdb24 dan http://localhost:6001/api/kwh/sdb25

Setelah *backend* aktif, *dashboard frontend* dijalankan melalui *script frontend* yang dibangun menggunakan React.js. Tampilan *dashboard* menampilkan data kWh dari tiap panel dalam bentuk grafik batang, yang terhubung secara *real-time* ke masing-masing API.

Secara keseluruhan, implementasi sistem berjalan dengan baik. Data dari panel SDB berhasil ditampilkan secara *real-time*, disimpan ke dalam *database*, dan divisualisasikan dengan rapi melalui *dashboard*. Sistem ini diharapkan dapat membantu proses monitoring konsumsi listrik di PT Astra Otoparts secara efisien dan informatif.

C. Hasil Pengujian

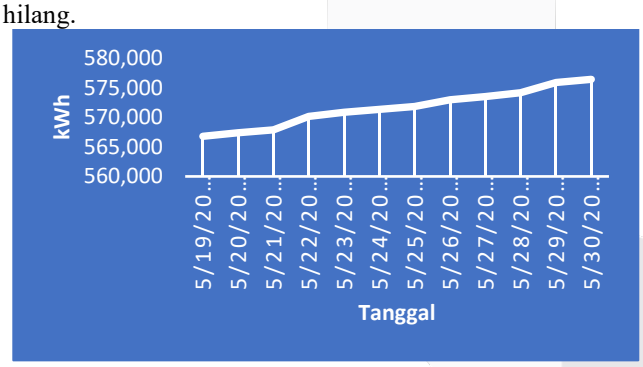
Pengujian dilakukan untuk memastikan semua komponen dalam sistem berjalan dengan baik dan sesuai dengan perancangan awal. Pengujian hanya dilakukan pada empat panel SDB, yaitu SDB 1, SDB 2, SDB 24, dan SDB 25. Panel tersebut terhubung ke perangkat kWh meter melalui protokol Modbus TCP/IP.

Pengujian dimulai dengan menghubungkan sistem ke jaringan lokal menggunakan Wi-Fi atau kabel LAN.

Dari hasil pengujian, kabel LAN memberikan koneksi yang lebih stabil dibandingkan Wi-Fi, karena koneksi Wi-Fi sering menyebabkan gangguan seperti *lag* atau *reconnect* pada protokol Modbus TCP/IP.

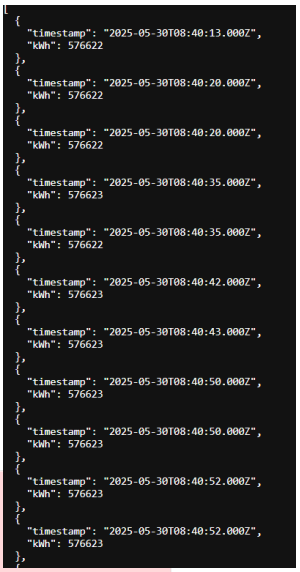
Setelah koneksi terbentuk, Node-RED dijalankan dan semua flow dikonfigurasi sesuai dengan panel masing-masing.

Data kWh dari perangkat berhasil dibaca secara terus-menerus setiap beberapa detik, kemudian secara otomatis disimpan ke dalam tabel di *database* MySQL yang berbeda untuk setiap panel. Untuk memastikan data tersimpan dengan benar, dilakukan pengecekan menggunakan aplikasi DBeaver. Hasilnya menunjukkan bahwa data kWh masuk ke setiap tabel secara *real-time* tanpa ada data yang rusak atau hilang.



GAMBAR 10
Database Mysql SDB 1

Setelah memastikan bahwa data tersimpan dengan baik, tahap selanjutnya adalah pengujian API yang dibangun pada folder *script backend*. Setiap *endpoint* diuji menggunakan *browser*. Hasilnya, semua *endpoint* dapat menampilkan data terbaru sesuai isi dari *database*, termasuk nilai total kWh dan tanggal-nya.



GAMBAR 11
Tampilan API

KWh yang diambil dari setiap panel SDB dikonfigurasi agar dapat terbaca secara otomatis setiap detik melalui Node-RED. Data tersebut diambil dengan menggunakan node Modbus Read dengan interval pembacaan yang telah ditentukan, sehingga data dikirim secara teratur dan konsisten ke dalam *database*. Hasil pengujian menunjukkan bahwa data yang tercatat setiap detik pada *dashboard* sesuai dengan angka yang ditampilkan pada display kWh meter fisik, sehingga dapat disimpulkan bahwa proses pengambilan data berjalan akurat tanpa terjadi keterlambatan atau kehilangan paket data.

Langkah akhir pengujian dilakukan di sisi *frontend*. Setelah *frontend* dijalankan, *dashboard* berhasil menampilkan grafik batang untuk masing-masing SDB. Grafik ini memvisualisasikan data kWh dari empat *endpoint* API dan dapat diakses secara lokal melalui *localhost*. Pengujian menunjukkan bahwa visualisasi berjalan lancar, data terupdate secara berkala, serta antarmuka dapat diakses tanpa adanya kesalahan. Secara keseluruhan, sistem telah lulus pengujian dari segi akurasi pembacaan, penyimpanan, dan penyajian data. Hasil ini menunjukkan bahwa *dashboard* monitoring yang telah dirancang siap digunakan untuk mendukung pemantauan konsumsi energi listrik produksi di PT Astra Otoparts.

V. KESIMPULAN

Berdasarkan hasil dari proses perancangan, implementasi, dan pengujian yang telah dilakukan, dapat disimpulkan bahwa sistem yang dikembangkan mampu melakukan pemantauan konsumsi energi listrik secara real-time menggunakan platform Node-RED dengan protokol komunikasi Modbus TCP/IP. Data konsumsi listrik dari kWh meter berhasil dikumpulkan dan disimpan ke dalam database MySQL untuk keperluan analisis dan pelaporan historis. Dashboard monitoring berbasis web yang dibangun mampu menampilkan data dari empat panel Sub Distribution Board (SDB 1, 2, 24, dan 25) dalam bentuk grafik batang yang informatif, sehingga memudahkan pengguna dalam memahami pola konsumsi energi secara efisien. Selain itu, backend API yang dikembangkan menggunakan Node.js dan Express.js terbukti berjalan dengan baik dalam menyajikan data dari database ke tampilan frontend. Hasil pengujian

sistem menunjukkan bahwa seluruh komponen bekerja secara stabil dan data yang ditampilkan konsisten dengan data yang tercatat di database. Kendala teknis seperti ketidakstabilan koneksi saat menggunakan jaringan Wi-Fi dapat diatasi dengan penggunaan kabel LAN, yang memberikan kestabilan komunikasi antar perangkat dalam sistem.

REFERENSI

[1] Nitol Saha;Md Masruk Aulia, Md. Mostafizur Rahman, Mohammed Shafiul Alam Khan, "IoT-Driven Cloud-based Energy and Environment Monitoring System for Manufacturing Industry," arXiv Preprint, pp. 1-6, 2024.

[2] S. Suwanda, "Application of the Internet of Things (IoT) in Production Management to Increase Production Efficiency in the Digital Era," Jurnal Ekonomi, vol. 13, no. 03, p. 1089–1100, 2024.

[3] A. G. P. M. M. D. Maman Abdurohman, "A Robust Internet of Things-Based Aquarium Control System Using Decision Tree Regression Algorithm," IEEE Access, vol. 10, p. 56937–56949, 2022.

[4] W. P. M. Fransiscus Xaverius Ariwibisono, "Implementasi Sistem Monitoring Produksi Energi PLTS Berbasis Protokol Modbus RTU dan Modbus TCP," Nuansa Informatika, vol. 17, no. 2, p. 109–118, 2023.

[5] E. S. W. D. Millenia Zwi Sabatina Sirait, "Kontrol Prototipe Ruang Monitoring Kesehatan Berbasis Node-RED," Jurnal Teknik Elektro dan Komputer TRIAC, vol. 9, no. 2, p. 135–140, 2022.

[6] BIF, Telkom University –, "Apa Itu MySQL? Pengertian MySQL, Cara Kerja, dan Kelebihannya," Telkom University – BIF, 2024. [Online]. Available: https://bif.telkomuniversity.ac.id/apa-itu-mysql/. [Accessed 09 Juni 2025].

[7] A. H. B. a. A. P. K. F. F. Nursaid, "Pengembangan Sistem Informasi Pengelolaan Persediaan Barang Dengan ReactJS Dan React Native Menggunakan Prototype (Studi Kasus : Toko Uda Fajri)," J-Ptiik.Ub.Ac.Id, vol. 4, pp. 46-55, 2022.

[8] I. Tahyudin and Zidni Iman Sholihati, "Pengembangan Aplikasi Tiga-Tingkat Menggunakan Metode Scrum pada Aplikasi Presensi Karyawan Glints Academy," Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), vol. 6, no. 10.2920, p. 169–176, 2022.

[9] W. Sipatuhar, "Perangkat Keras Komputer," 2020.

[10] J. W. I. P. a. M. E. A. Syah, "Pengenalan Perangkat Lunak (software) Pada Komputer," 2020.