

Desain Sistem Kontrol *Container Station* Menggunakan Metode *Waterfall* Berbasis *Internet of Things (IoT)*

Yuliad Erdani^{1✉}, Rizqi Aji Pratama², Ginna Intan Fadila³, Bahdin Ahad Badia⁴

^{1,2,3} Teknik Otomasi Manufaktur dan Mekatronika, Politeknik Manufaktur Bandung, Indonesia

⁴ Prodi Teknologi Metalurgi Industri Logam, Politeknik Tridaya Virtu Morosi, Indonesia

Informasi Artikel

Riwayat Artikel

Diserahkan : 07-05-2024

Direvisi : 19-05-2024

Diterima : 20-05-2024

Kata Kunci:

*Blynk; Internet of Things;
MQTT;
NodeMCU; Waterfall*

Keywords :

*Blynk; Internet of Things;
MQTT;
NodeMCU; Waterfall*

ABSTRAK

Industri logistik dan manajemen *container* menghadapi tantangan signifikan dalam pengelolaan *container station*, seperti kompleksitas operasional, risiko keamanan, dan kurangnya visibilitas. Hal ini dapat menyebabkan kerugian ekonomi yang besar akibat kehilangan atau kerusakan barang, keterlambatan pengiriman, serta pelanggaran keamanan. Penelitian ini bertujuan merancang sistem kontrol *container station* menggunakan metode *waterfall* berbasis *internet of things (IoT)*. Metode *waterfall* dipilih untuk pendekatan sistematis dan terstruktur, sementara IoT memungkinkan pemantauan dan pengendalian jarak jauh secara *real-time*. Metode *waterfall* digunakan untuk memastikan pengembangan sistem dilakukan secara berurutan dari analisis kebutuhan, system desain, implementasi, pengujian, hingga pemeliharaan sistem. Pengiriman data pada sistem ini dilakukan menggunakan protokol *Message Queuing Telemetry Transport (MQTT)*. Hasil pengujian menunjukkan integrasi antara perangkat *Programmable Logic Controller* dan jaringan *Internet of Things* sukses terwujud melalui *NodeMCU* sebagai penghubung. Aplikasi *Blynk* digunakan untuk kontrol dan pemantauan *container station*. Dengan aplikasi ini, pengguna dapat dengan mudah mengontrol dan memantau *container station* melalui *push button* dan *Light Emission Diode (LED)*.

ABSTRACT

The logistics and container management industry faces significant challenges in the management of container stations, such as operational complexity, security risks, and lack of visibility. This can lead to substantial economic losses due to loss or damage of goods, shipping delays, and security breaches. This research aims to design an efficient and integrated container station control system using the waterfall method based on the Internet of Things (IoT). The waterfall method is chosen for a systematic and structured approach, while IoT enables real-time remote monitoring and control. The waterfall method is used to ensure that system development is carried out sequentially from requirements analysis, system design, implementation, testing, to system maintenance. Data transmission in this system is performed using the Message Queuing Telemetry Transport (MQTT) protocol. The test results show that the integration between the Programmable Logic Controller devices and the Internet of Things network has been successfully achieved through NodeMCU as the connector. The Blynk application is used for container station control and monitoring. With this application, users can easily control and monitor the container station through push buttons and Light Emission Diodes(LED).

Corresponding Author :

Yuliadi Erdani

Teknik Otomasi Manufaktur dan Mekatronika, Politeknik Manufaktur Bandung, Indonesia

Jl. Kanayakan No. 21 Dago, Kecamatan Comblong, Kota Bandung, Jawa Barat 40135

Email: yul_erdani@yahoo.com

PENDAHULUAN

Seiring dengan pertumbuhan perdagangan global, industri logistik dan manajemen container menghadapi tantangan yang semakin kompleks. Proses penanganan container melibatkan serangkaian aktivitas yang kompleks, seperti pemindahan, penyimpanan, pemeriksaan, dan pengawasan (Ramadhan et al., 2018). Kesalahan atau inefisiensi dalam pengelolaan *container station* dapat mengakibatkan kerugian ekonomi yang signifikan, seperti kehilangan atau kerusakan barang, keterlambatan pengiriman, serta pelanggaran keamanan (Rodrigues et al., 2020). Oleh karena itu, dibutuhkan solusi yang efektif dan terintegrasi untuk meningkatkan efisiensi, keamanan, dan visibilitas dalam proses penanganan container.

Perkembangan teknologi *Internet of Things (IoT)* telah membuka peluang baru dalam mengatasi tantangan tersebut. *Internet of Things (IoT)* memungkinkan perangkat dan sistem untuk saling terhubung dan bertukar data secara real-time melalui jaringan internet (Farooq et al., 2015). Dengan memanfaatkan *Internet of Things (IoT)*, perusahaan logistik dapat memantau lokasi, kondisi, dan pergerakan container secara akurat, serta mengoptimalkan alokasi sumber daya dan meningkatkan keamanan (Alvi et al., 2021). Namun, implementasi teknologi *Internet of Things (IoT)* dalam sistem kontrol *container station* memerlukan pendekatan yang sistematis dan terstruktur untuk memastikan keberhasilan implementasi.

Industri logistik dan manajemen container memainkan peran penting dalam rantai pasokan global. Namun, terdapat tantangan yang signifikan dalam pengelolaan *container station*, seperti kompleksitas operasional, risiko keamanan, dan kurangnya visibilitas dalam proses penanganan container (Rodrigues et al., 2020). Dengan berkembangnya teknologi *Internet of Things (IoT)*, terbuka peluang untuk mengintegrasikan solusi yang lebih efisien dan terpadu dalam mengatasi tantangan tersebut.

Container station merupakan fasilitas yang digunakan untuk menyimpan dan menangani container dalam jangka waktu tertentu sebelum dikirimkan ke lokasi tujuan. Proses penanganan container melibatkan serangkaian aktivitas kompleks, seperti pemindahan, penyimpanan, pemeriksaan, dan pengawasan. Kesalahan dalam pengelolaan container dapat mengakibatkan kerugian ekonomi yang besar, seperti kehilangan atau kerusakan barang, keterlambatan pengiriman, dan pelanggaran keamanan (Ramadhan et al., 2018).

Untuk meningkatkan efisiensi dan keamanan dalam pengelolaan *container station*, beberapa perusahaan logistik di Indonesia mulai mengadopsi teknologi *Internet of Things (IoT)*. *Internet of Things (IoT)* memungkinkan perangkat dan sistem untuk saling terhubung dan bertukar data secara *real-time* melalui jaringan internet (Ashari & Abbas, 2020). Dengan memanfaatkan *Internet of Things (IoT)*, perusahaan logistik dapat memantau lokasi, kondisi, dan pergerakan container secara akurat, serta mengoptimalkan alokasi sumber daya dan meningkatkan keamanan (Kusnandar et al., 2021).

Dalam upaya meningkatkan efisiensi dan keamanan pengelolaan *container station*, banyak perusahaan logistik mulai mengadopsi teknologi *Internet of Things (IoT)*. Teknologi ini memungkinkan pemantauan dan pengendalian jarak jauh secara real-time melalui perangkat yang terhubung ke internet (Farooq et al., 2015). Dengan memanfaatkan IoT, perusahaan dapat memantau lokasi, kondisi, dan pergerakan container secara akurat, serta mengoptimalkan alokasi sumber daya dan meningkatkan keamanan.

Metode *waterfall* merupakan salah satu model pengembangan perangkat lunak yang bersifat linear dan sekuensial, dengan setiap tahapan harus diselesaikan sebelum melanjutkan ke tahap berikutnya (Larman & Basili, 2023). Metode ini dipilih karena sifatnya yang terstruktur dan sistematis, sehingga memudahkan proses pengembangan dan implementasi sistem. Dengan mengombinasikan metode *waterfall* dan teknologi *Internet of Things (IoT)*, diharapkan dapat dirancang sebuah sistem kontrol *container station* yang efisien, terintegrasi, dan mampu meningkatkan visibilitas serta keamanan dalam penanganan container.

Namun, implementasi teknologi *Internet of Things (IoT)* dalam sistem kontrol *container station* memerlukan pendekatan yang sistematis dan terstruktur. Oleh karena itu, penelitian ini mengusulkan penggunaan metode *waterfall* dalam merancang sistem kontrol *container station* berbasis *Internet of Things (IoT)*. Metode *waterfall* merupakan model pengembangan perangkat lunak yang bersifat linear dan sekuensial, dengan setiap tahapan harus diselesaikan sebelum

melanjutkan ke tahap berikutnya (Larman & Basili, 2023). Metode ini dipilih karena sifatnya yang terstruktur dan sistematis, sehingga memudahkan proses pengembangan dan implementasi sistem.

Penelitian sebelumnya telah mengeksplorasi pemanfaatan *Internet of Things (IoT)* dalam manajemen container, namun sebagian besar masih terbatas pada aspek tertentu seperti pemantauan lokasi (Ashari & Abbas, 2020; Aziz et al., 2019) atau kondisi lingkungan (Singh & Gupta, 2020). Penelitian ini bertujuan untuk merancang sebuah sistem kontrol *container station* yang komprehensif dan terintegrasi, mencakup aspek pemantauan, pengendalian, keamanan, dan optimalisasi sumber daya dengan menggunakan metode *waterfall* dan teknologi *Internet of Things (IoT)*.

Dengan menggabungkan metode *waterfall* dan teknologi *Internet of Things (IoT)*, sistem yang dirancang diharapkan dapat meningkatkan efisiensi operasional, meminimalkan kesalahan manusia, dan meningkatkan keamanan serta visibilitas dalam penanganan container. Selain itu, penelitian ini juga diharapkan dapat memberikan kontribusi dalam pengembangan solusi *Internet of Things (IoT)* di bidang logistik dan manajemen rantai pasokan, serta mendukung upaya keberlanjutan melalui penghematan sumber daya dan pengurangan jejak karbon (Singh & Gupta, 2020).

Beberapa penelitian terdahulu telah mencoba mengeksplorasi solusi berbasis *Internet of Things (IoT)* dalam pengelolaan *Container Station*. Rodriguez et al. (2019) mengusulkan sebuah sistem pemantauan kontainer yang mengintegrasikan sensor, gateway, dan cloud computing, namun masih terbatas pada fungsi pemantauan saja. Sementara itu, Liang et al. (2020) mengembangkan model optimasi penyusunan kontainer menggunakan algoritma heuristik, tetapi belum mengintegrasikan aspek pemantauan dan kontrol secara *real-time*.

Dalam penelitian lain, Mejjaoui & Babiceanu, 2018) mengembangkan sebuah sistem manajemen kontainer berbasis *Internet of Things (IoT)* dengan menggunakan teknologi RFID dan komunikasi nirkabel. Sistem ini mampu memantau lokasi dan status kontainer secara *real-time*, namun belum mencakup aspek kontrol dan otomatisasi proses. Di sisi lain, Cao et al. (2021) mengusulkan sebuah kerangka kerja untuk pengelolaan *Container Station* yang terintegrasi dengan teknologi *blockchain* dan *Internet of Things (IoT)*, dengan fokus utama pada aspek keamanan dan pelacakan.

Meskipun dari beberapa penelitian tersebut telah memberikan kontribusi berharga dalam mengeksplorasi potensi *Internet of Things (IoT)* dalam pengelolaan logistik kontainer, masih terdapat ruang untuk pengembangan lebih lanjut dalam merancang sistem yang terintegrasi dan efisien yang mencakup aspek pemantauan, kontrol, dan otomatisasi proses secara menyeluruh.

Penelitian ini bertujuan untuk merancang sebuah sistem kontrol *container station* yang komprehensif dan terintegrasi dengan menggunakan metode *waterfall* dan teknologi *Internet of Things (IoT)*. Dengan menggabungkan keduanya, sistem yang dirancang diharapkan dapat meningkatkan efisiensi operasional, meminimalkan kesalahan manusia, dan meningkatkan keamanan serta visibilitas dalam penanganan container di Indonesia. Penelitian ini juga diharapkan dapat memberikan kontribusi dalam pengembangan solusi *Internet of Things (IoT)* di bidang logistik dan manajemen rantai pasokan di Indonesia.

Berdasarkan penelitian yang dilakukan, ditemukan bahwa *Programmable Logic Controller (PLC)* tidak dapat terhubung secara langsung atau diintegrasikan ke dalam sistem *Internet of Things (IoT)* tanpa bantuan perangkat tambahan. Untuk mengatasi hal ini, penggunaan mikrokontroler sebagai perantara antara *Programmable Logic Controller (PLC)* dan internet diperlukan. Dalam konteks pengontrolan dan pemantauan *container station*, penulis memutuskan untuk mengembangkan sebuah aplikasi menggunakan platform *Blynk*. Aplikasi tersebut bertujuan untuk mengontrol dan memantau *Programmable Logic Controller (PLC)* dengan memanfaatkan protokol komunikasi *Message Queuing Telemetry Transport (MQTT)*.

METODE PENELITIAN

Penelitian ini menggunakan pendekatan penelitian dan pengembangan (*Research and Development*) untuk merancang dan mengimplementasikan sistem kontrol *Container Station* berbasis *Internet of Things (IoT)* dengan metode *Waterfall*. Metode penelitian yang digunakan dapat dijelaskan sebagai berikut:

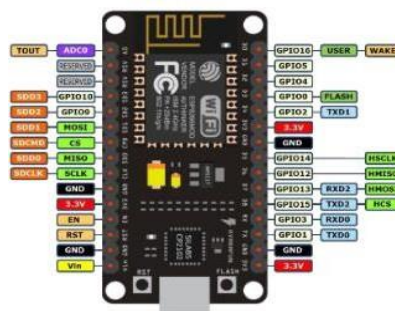
1. Studi Literatur

Studi Literatur Pada tahap ini, dilakukan studi literatur yang komprehensif untuk mempelajari konsep-konsep terkait, seperti *Internet of Things* (IoT), sistem kontrol, dan pengelolaan *Container Station*. Studi literatur juga mencakup tinjauan terhadap penelitian-penelitian sebelumnya yang relevan untuk memahami *state-of-the-art* dan mengidentifikasi gap penelitian.

2. Analisis Kebutuhan

Analisis Kebutuhan Tahap ini melibatkan pengumpulan dan analisis kebutuhan fungsional dan non-fungsional dari sistem kontrol *Container Station* berbasis *Internet of Things* (IoT). Kegiatan yang dilakukan meliputi wawancara dengan pemangku kepentingan (seperti operator *Container Station*, perusahaan logistik, dan pelanggan), observasi lapangan, serta analisis proses bisnis yang ada.

a. NNodeMCUESP8266



Gambar 1. Arsitektur *NodeMCUESP8266*

NodeMCU ESP8266 merupakan sebuah modul yang meletakkan NodeMCU dan ESP8266 pada satu tempat yang sama sehingga pengguna tidak perlu lagi untuk membeli keduanya secara terpisah maupun merangkainya (Gunawan et al., 2020). Pada sistem ini, NodeMCU digunakan sebagai kontroler yang mengumpulkan data perhitungan lalu mengirimkan data tersebut ke MongoDB menggunakan protokol *Message Queuing Telemetry Transport* (MQTT).

b. Sensor*ProximityE18-D80NK*



Gambar 2. Sensor Infra Merah Tipe E18-D80K

Sensor infra merah dapat mendeteksi suatu benda tanpa harus bersentuhan dengan benda tersebut (Rahayu et al., 2021). Sensor ini digunakan sebagai pendeteksi dan penghitung jumlah botol pada *container station*.

c. *ArduinoIDE*

IDE Arduino merupakan sebuah editor yang digunakan untuk menulis, mengompilasi, dan mengunggah program ke dalam papan Arduino (Efendi et al., 2023). Lingkungan pengembangan *Arduino IDE* terdiri dari editor teks untuk menulis kode, area

pesan, konsol teks, toolbar dengan fungsi umum, serta berbagai opsi menu lainnya (Santoso et al., 2021). Digunakan untuk membuat *sketch* program perhitungan botol.

d. KEPServerEX

KEPServerEX merupakan salah satu jenis OPC (OLE for Process Control) yang banyak digunakan dalam dunia industri. OPC merupakan sebuah platform komunikasi yang dapat menerima data dari PLC, DCS, mikrokontroler, dan perangkat lainnya (R. Priyatna & A. Andang, 2021). *KEPServerEX* berperan penting sebagai platform server komunikasi yang memiliki kemampuan untuk menari data I/O dari perangkat yang terhubung. *KEPServerEX* digunakan sebagai penghubung *Programmable Logic Controller* (PLC) dengan *Message Queuing Telemetry Transport* (MQTT).

e. Blynk

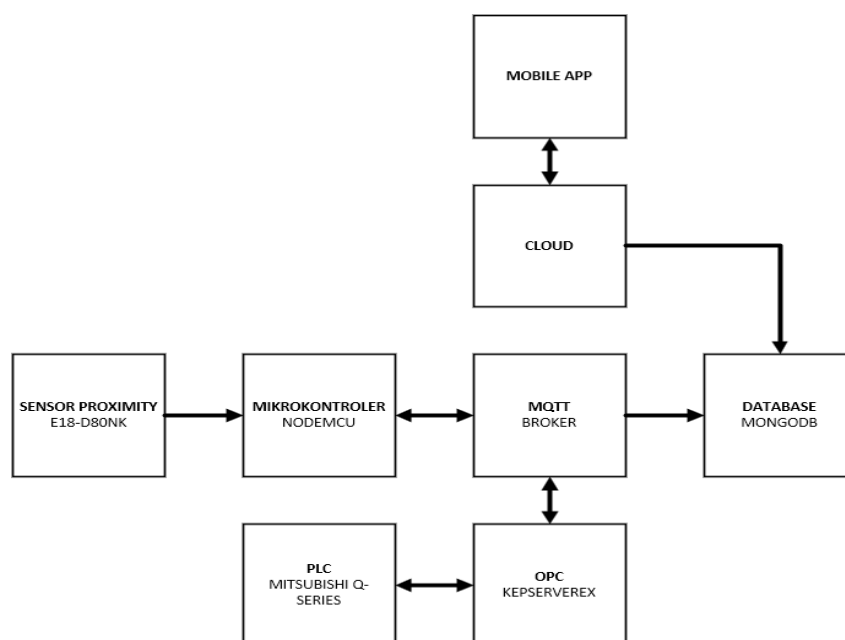
Blynk adalah sebuah aplikasi ponsel cerdas yang dirancang untuk menghubungkan dan mengendalikan berbagai perangkat berbasis *Internet of Things* melalui jaringan internet (I. Pramudya et al., 2020). Pada sistem ini, *Blynk* digunakan sebagai *framework* pembuatan desain antarmuka pengguna untuk kendali dan *monitoring container station*.

f. MongoDB

MongoDB merupakan sistem basis data dokumen yang mengadopsi konsep NoSQL (*Not Only SQL*). MongoDB tidak menggunakan tabel, kolom, dan baris, melainkan hanya menggunakan koleksi dan dokumen (Kurniawanto Putra et al., 2016). MongoDB digunakan untuk menyimpan, mengelola, dan mengakses data.

3. Desain Sistem

Desain Sistem Berdasarkan hasil analisis kebutuhan, tahap selanjutnya adalah merancang arsitektur sistem yang mencakup komponen perangkat keras dan perangkat lunak. Pada tahap ini, dilakukan pemilihan teknologi *Internet of Things* (IoT) yang sesuai, seperti sensor, aktuator, dan modul komunikasi, serta perancangan antarmuka pengguna yang *user-friendly*.



Gambar 3. Gambaran Umum Desain Sistem Kontrol

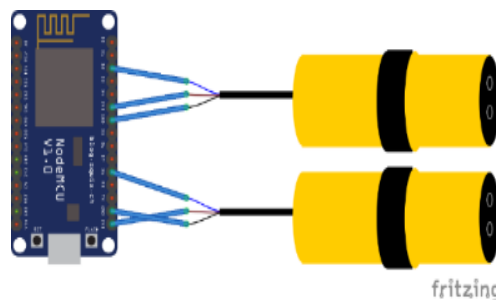
Gambar 3 menjelaskan gambaran umum sistem kendali dan monitoring yang akan diterapkan pada *container station* berbasis *internet of things*. Sensor infra merah akan mendeteksi dan menghitung botol, kemudian NodeMCU akan mengendalikan pengiriman data

perhitungan ke *database*. Sedangkan KEPServerEX akan mengambil data dari *Programmable Logic Controller* (PLC) lalu mengirimkan data tersebut melalui fitur IoT gateway. Data-data tersebut akan dikirimkan ke *database* MongoDB melalui *Message Queuing Telemetry Transport (MQTT) broker* untuk disimpan dan dikelola (Defi et al., 2024).

4. Implementasi Sistem

Implementasi Sistem Setelah desain sistem selesai, dilanjutkan dengan implementasi sistem menggunakan metode *Waterfall*. Tahap ini merupakan tahap pemrograman yang dilakukan untuk menghitung botol yang keluar menuju *conveyor*, penyimpanan data pada MongoDB, serta program *user interface* menggunakan *Blynk*.

a. Perhitungan Botol



Gambar 4. Rancangan Rangkaian

Gambar 4 menjelaskan rancangan rangkaian sensor infra merah E18-D80NK yang terhubung dengan NodeMCU. Untuk pembuatan *sketch* program pada aplikasi Arduino IDE seperti pada gambar 5. Program di bawah ini menggunakan bahasa pemrograman Arduino untuk platform ESP8266 yang berfungsi sebagai penghubung ESP8266 ke *jaringan wireless fidelity (WiFi)* dan mengirimkan data jumlah botol hitam dan putih melalui protokol komunikasi *Message Queuing Telemetry Transport (MQTT)* ke *Message Queuing Telemetry Transport (MQTT) broker*. Program ini juga mengatur pembacaan sensor untuk menghitung jumlah botol berwarna putih maupun hitam berdasarkan perubahan nilai sensor secara berulang dengan interval delay 1 detik.

```
esp_mong
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <ArduinoJson.h>

// Deklarasi pin HITAM
const int HITAM = 4;
const int PUTIH = 5;
int L;
int M;
// inisialisasi variabel
int hitungHitam = 0;
int hitungPutih = 0;
int kondisi1 = 0;
int kondisi2 = 0;
```

Gambar 5. Program Penghitung botol hitam dan putih

b. Penyimpanan Data pada MongoDB



```
JS app3.js X
JS app3.js > | DataSchema
1  const mqtt = require('mqtt');
2  const mongoose = require('mongoose');
3
4  const brokerUrl = 'mqtt://91.121.93.94';
5  const brokerPort = 1883;
6  const topic = 'Nicetry/msg';
7
8  const dbUrl = 'mongodb+srv://ginfad12:YukBisaYuk12@mydb.1g64bgy.mongodb.net/test';
9  const dbName = 'try';
10 const collectionName = 'hitungbarang';
```

Gambar 6. Program Penyimpanan Data pada MongoDB

Gambar 5 dan gambar 6 merupakan program JavaScript untuk menyimpan data dari protocol *Message Queuing Telemetry Transport (MQTT)* ke database MongoDB dengan menggunakan library “mqtt” dan “mongoose”. Ketika pesan diterima oleh topic *Message Queuing Telemetry Transport (MQTT)*, program akan mengambil nilai pesan dan mengubahnya menjadi format JSON dan memvalidasi nilai tersebut. Jika nilai *valid*, data akan disimpan ke database.

5. Pengujian dan Evaluasi

Setelah implementasi sistem selesai, dilakukan serangkaian pengujian untuk memvalidasi kinerja sistem. Pengujian meliputi aspek fungsionalitas, keamanan, performa, dan usabilitas. Selain itu, dilakukan evaluasi terhadap dampak sistem terhadap efisiensi operasional, akurasi data, dan visibilitas pergerakan kontainer di *Container Station*.

6. Analisis dan Pelaporan

Tahap terakhir adalah menganalisis hasil pengujian dan evaluasi, serta menyusun laporan penelitian yang mencakup temuan, diskusi, dan rekomendasi untuk penelitian dan pengembangan lebih lanjut.

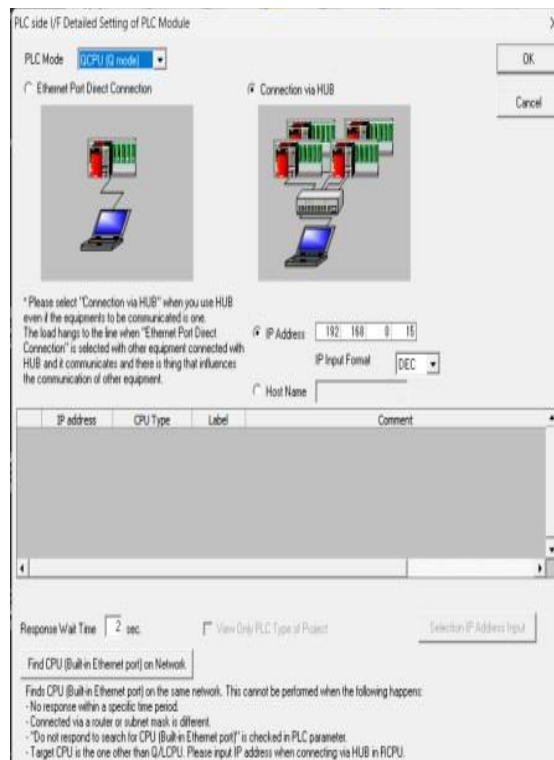
HASIL DAN PEMBAHASAN

Berdasarkan analisis kebutuhan dan desain sistem yang telah dilakukan, dihasilkan sebuah sistem kontrol *Container Station* berbasis *Internet of Things (IoT)* dengan arsitektur sebagai berikut:

A. Integrasi PLC dengan IoT Menggunakan NodeMCU

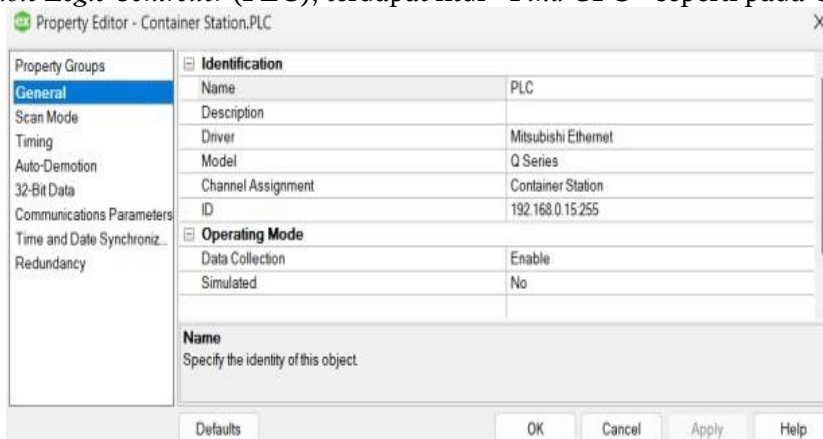
Integrasi *Programmable Logic Controller (PLC)* dengan *Internet of Things (IoT)* menggunakan NodeMCU adalah salah satu pendekatan yang digunakan untuk menghubungkan perangkat *Programmable Logic Controller (PLC)* dengan system *Internet of Things (IoT)*. Dengan menggunakan NodeMCU, perangkat *Programmable Logic Controller (PLC)* dapat mengirimkan dan menerima data secara *real-time* melalui koneksi internet.

1. Konfigurasi Komuniaksi PLC dengan KEPServerEX



Gambar 7. Konfigurasi Programmable Logic Controller (PLC)

Tipe *Programmable Logic Controller (PLC) Q-Series* dipilih sesuai dengan *Programmable Logic Controller (PLC)* yang digunakan pada *container station*. Konfigurasi *Programmable Logic Controller (PLC)* pada GX Works2 dimulai dari pengaturan perangkat seperti modul I/O dan perangkat tambahan lainnya. Pengaturan ini juga menentukan parameter perangkat keras seperti alamat memori atau jenis komunikasi dan bahasa pemrograman yang digunakan. Untuk mengetahui IP address *Programmable Logic Controller (PLC)*, terdapat fitur “*Find CPU*” seperti pada Gambar 7.



Gambar 8. Konfigurasi PLC pada KEPServerEX

Data *input* dan *output* akan dikumpulkan pada OPC KEPServerEX sehingga pemilihan *driver* harus sesuai dengan yang digunakan dalam sistem agar pertukaran informasi dapat berjalan dengan baik. Selain itu, pemilihan model serta pengaturan parameter komunikasi seperti IP address, nomor *port*, dan protokol komunikasi juga harus diperhatikan seperti pada Gambar 8.

KEPServerEX menyediakan fitur *Internet of Things (IoT) Gateway* yang mendukung protokol komunikasi *Message Queuing Telemetry Transport (MQTT)* dalam implementasi *Internet of*

Things (IoT). Untuk menggunakan fitur ini, pengguna perlu memasukkan informasi mengenai *client*, *message*, dan *subscriptions* yang ditampilkan pada gambar 9.



Gambar 9. Konfigurasi *Client Message Queuing Telemetry Transport (MQTT)*

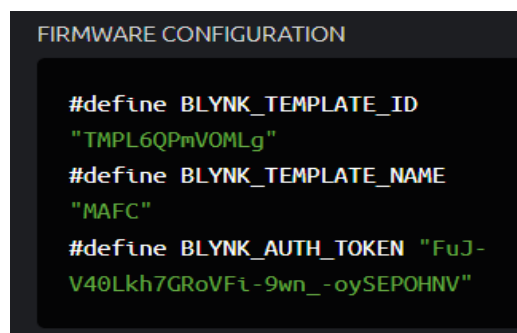
2. Konfigurasi MQTT pada NODEMCU

Konfigurasi MQTT pada NodeMCU dimulai dengan NodeMCU yang terhubung ke jaringan WiFi yang sesuai. Setelah terhubung, koneksi dengan broker MQTT dibuat melalui pengaturan alamat IP dan port yang sesuai dengan broker yang akan digunakan. Dalam konfigurasi ini, topik MQTT ditentukan untuk memfasilitasi pengiriman dan penerimaan pesan antara NodeMCU dan broker. NodeMCU kemudian dikonfigurasi agar dapat mempublikasikan pesan yang relevan ke topik MQTT yang telah ditentukan. Dengan demikian, NodeMCU dapat berkomunikasi dengan broker dan menukar pesan sesuai dengan topik yang ditentukan.

B. Perangkat-perangkat Lunak

Perancangan perangkat lunak menggunakan Blynk dilakukan untuk menciptakan aplikasi interaktif yang menghubungkan perangkat IoT dengan pengguna. Sehingga, pengguna dapat mengontrol dan memantau perangkat secara *real-time* melalui aplikasi *mobile*

1. Konfigurasi Aplikasi Blynk



Gambar 10. *Auth Token* Proyek

Aplikasi Blynk digunakan sebagai aplikasi pengontrol perangkat IoT. Proyek dimulai dengan membuat proyek baru dan memilih tipe data, misalnya NodeMCU, sebagai perangkat target. Selanjutnya, pengaturan koneksi dilakukan melalui *WiFi* atau *Bluetooth*. Setiap proyek dalam Blynk memiliki auth token unik yang berfungsi sebagai kunci autentikasi untuk perangkat

terhubung dan berkomunikasi dengan server *Blynk*. Token ini kemudian dimasukkan ke dalam program *Arduino IDE*.

2. Kode Program *Blynk*

```
ESP8266_wifi_blynk_3_
#define BLYNK_PRINT Serial

#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <BlynkSimpleEsp8266.h>

#define BLYNK_TEMPLATE_ID          "TMPL6QFmVOMLg"
#define BLYNK_TEMPLATE_NAME        "MAFC"
#define BLYNK_AUTH_TOKEN           "FuJ-V40Lkh7GRoVFi-9wn_-oySEPOHNV"

// Define virtual pin
#define Start V0
#define Stop V1
#define Reset V2
#define Return V3
#define Emergency_Stop V4
#define Cylinder_1 V5
#define Cylinder_2 V6
#define Run V7
#define Stand_By V8
#define Sensor V9
#define AUTOMANUAL V10
#define RED_LED V11

char ssid[] = "Ginna I";
char pass[] = "123ginfadddd";

const char* mqtt_broker = "91.121.93.94";
const int mqtt_port = 1883;
const char* mqtt_topic = "Nicetry/Input/Start";
```

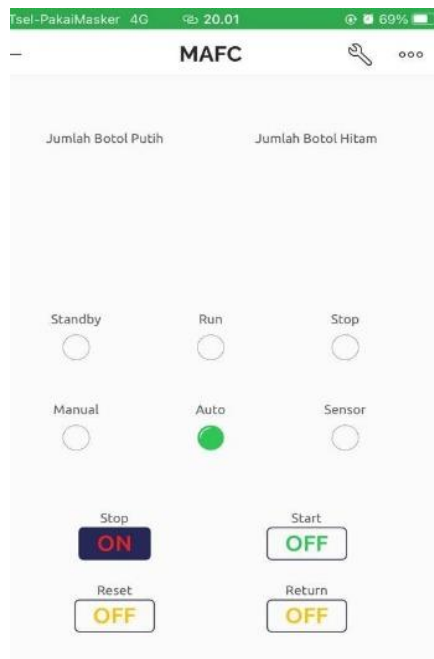
Gambar 11. Kode Program *Blynk*

Kode program ini dirancang untuk menghubungkan aplikasi *Blynk* dengan perangkat PLC menggunakan NodeMCUESP8266. Konfigurasi seperti templateID, nama template, token autentikasi, dan pin virtual digunakan untuk menghubungkan proyek dengan *Blynk* dan mengontrol perangkat. Proses inisialisasi *WiFi* dan koneksi MQTT memungkinkan perangkat terhubung dengan server *Blynk* dan melakukan pertukaran pesan secara *real-time*. Fungsi *setup()* melakukan pengaturan awal, sementara *loop()* menjalankan komunikasi dengan server *Blynk* dan broker MQTT. Fungsi *BLYNK_WRITE* merespons perubahan pada pin virtual dan mengirimkan pesan MQTT yang sesuai. Program ini memungkinkan pengontrolan dan pemantauan perangkat melalui aplikasi *Blynk* yang telah dibuat dengan menggunakan perangkat ESP8266.

3. Tampilan Antar Muka Pengguna

Pada Gambar 12 terdapat beberapa item kontrol dan *monitoring* untuk mengetahui status *plant container station*. Item kontrol terdiri dari empat *push button* yaitu *start*, *stop*, *restart*, dan *return*. Sementara itu, pada bagian *monitoring* terdiri dari enam lampu dan dua *value display* yang digunakan untuk menampilkan jumlah botol putih dan hitam.

Salah satu aspek penting dalam perancangan sistem kontrol *container station* berbasis IoT adalah tampilan antarmuka pengguna yang intuitif dan user-friendly. Antar muka pengguna berfungsi sebagai jembatan antara pengguna dan sistem, sehingga memudahkan interaksi dan memaksimalkan manfaat dari fitur-fitur yang tersedia (Muliawan et al., 2022).



Gambar 12. Tampilan Antarmuka Aplikasi

Halaman utama antarmuka pengguna dirancang untuk memberikan gambaran umum tentang kondisi *container station* secara real-time. Informasi yang ditampilkan meliputi jumlah *container* yang ada di lokasi, status setiap *container* (misalnya, dalam perjalanan, sedang diperiksa, atau siap dikirim), serta peringatan atau notifikasi penting terkait potensi risiko atau kondisi tidak normal (Ramadhan et al., 2018).

Antarmuka halaman utama juga dilengkapi dengan peta interaktif yang menampilkan lokasi fisik setiap container di area *container station*. Pengguna dapat mengklik pada setiap ikon container untuk mendapatkan informasi lebih rinci, seperti nomor identifikasi, tujuan pengiriman, kondisi container (suhu, kelembaban, getaran), dan riwayat perpindahan (Alvi et al., 2021).

Pemantauan dan Pengendalian Antarmuka pengguna juga menyediakan fitur pemantauan dan pengendalian yang memungkinkan pengguna untuk mengawasi dan mengelola aktivitas di *container station* secara *real-time*. Melalui antarmuka ini, pengguna dapat memantau kondisi lingkungan seperti suhu, kelembaban, dan getaran yang dapat mempengaruhi integritas barang dalam container (Singh & Gupta, 2020).

Selain itu, pengguna juga dapat melakukan pengendalian jarak jauh terhadap perangkat IoT, seperti membuka atau menutup pintu container, mengaktifkan sistem pendingin, atau menyalakan lampu peringatan. Fitur ini memungkinkan pengelolaan yang lebih efisien dan cepat tanggap terhadap situasi darurat atau kondisi tidak normal (Kusnandar et al., 2021).

Pengguna dapat mengatur periode waktu yang ingin dianalisis, memilih jenis data yang ingin ditampilkan, serta mengekspor laporan dalam berbagai format seperti PDF atau Excel untuk keperluan dokumentasi dan audit (Ashari & Abbas, 2020).

C. Pengujian Aplikasi

Serangkaian pengujian fungsional dan non-fungsional dilakukan untuk memastikan sistem berjalan dengan baik dan memenuhi kebutuhan yang telah ditetapkan. Pengujian meliputi aspek fungsionalitas, keamanan, performa, dan usability (Cao et al., 2021; Zohri et al., 2019). Pengujian dilakukan terhadap aplikasi yang telah dibuat. Berikut adalah *table* pengujian sistem kontrol dan indikator pada aplikasi.

Tabel 1. Pengujian Sistem Kontrol

NO.	Push Button	Keterangan
-----	-------------	------------

- | | | |
|---|---------------|---|
| 1 | <i>Start</i> | <i>Push button start</i> dapat membuat silinder bergerak maju untuk mengeluarkan botol putih. |
| 2 | <i>Stop</i> | <i>Push button stop</i> dapat membuat silinder bergerak maju untuk mengeluarkan botol hitam. |
| 3 | <i>Reset</i> | <i>Push button Reset</i> sebagai syarat setelah tombol <i>emergency stop</i> dilepaskan |
| 4 | <i>Return</i> | <i>Push button Reset</i> sebagai syarat setelah tombol <i>emergency stop</i> dilepaskan. |

Tabel 2. Pengujian Sistem Indikator

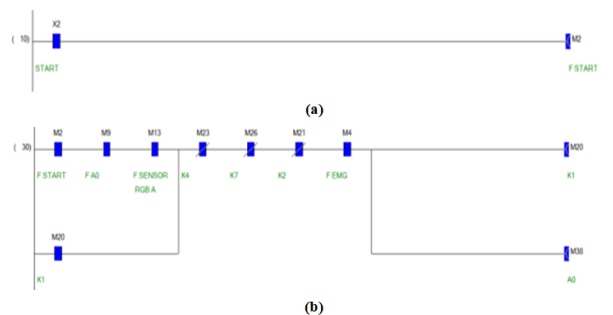
NO.	Indikator	Keterangan
1	<i>Run</i>	Indikator <i>run</i> akan menyala saat tombol <i>start</i> atau <i>stop</i> ditekan.
2	<i>Standby</i>	Indikator <i>standby</i> akan menyala saat <i>plant</i> berada pada posisi awal.
3	<i>Stop</i>	Indikator <i>stop</i> akan menyala saat <i>plant</i> mengaktifkan <i>emergency stop</i> .
4	<i>Auto</i>	Indikator <i>auto</i> akan menyala saat <i>plant</i> dalam mode otomatis.
5	<i>Manual</i>	Indikator <i>manual</i> akan menyala saat <i>plant</i> dalam mode manual.
6	<i>Sensor</i>	Indikator akan menyala saat sensor mendeteksi benda.

1. Pengujian Sistem Kontrol Push Button

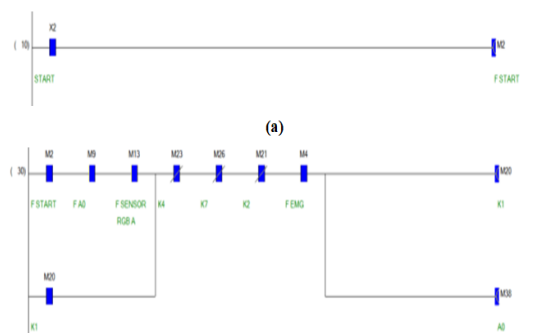


Gambar 13. Push Button Pada Aplikasi Blynk

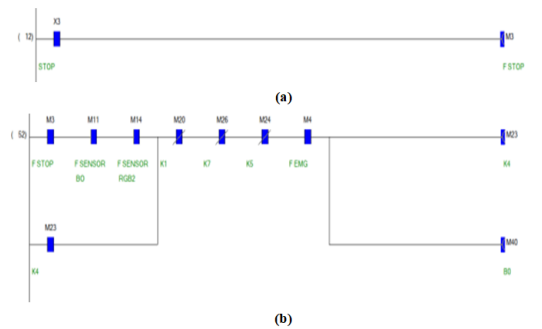
Pada gambar 13 ditampilkan push button untuk mengendalikan container station. Kontrol plant dapat dilakukan dengan menekan tombol yang tersedia pada aplikasi. Pada Gambar 14, gambar 15, dan gambar 16 menunjukkan bahwa saat *push button* aplikasi ditekan, maka data *Boolean* bernilai “true” akan dikirimkan dan mengaktifkan alamat memori yang sesuai dengan alamat yang dituju.



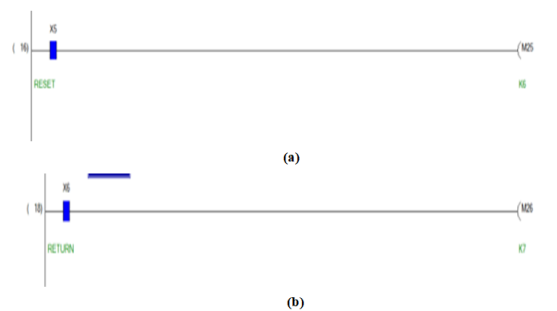
Gambar 14. Program PLC saat tombol *start* ditekan (a) Program PLC tombol *start* (b) Program PLC untuk silinder mendorong botol berwarna putih



Gambar 15. Program PLC saat tombol *start* ditekan (a) Program PLC tombol *start* (b) Program PLC untuk silinder mendorong botol berwarna putih



Gambar 16. Program PLC saat tombol stop ditekan (a) Program PLC tombol stop (b) Program PLC untuk silinder mendorong botol berwarna hitam



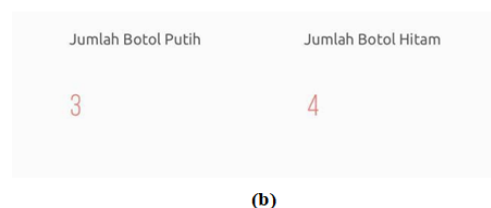
Gambar 17. *Push Button* (a) *Reset* (b) *Return* sebagai syarat untuk memulai kembali kendali *plant*

2. Pengujian Sistem Monitoring Indikator



Gambar 18. Indikator pada aplikasi *Blynk*

```
14:10:09.737 -> Jumlah botol hitam:
14:10:09.737 -> 4
14:10:09.737 -> Jumlah botol putih:
14:10:09.737 -> 3
(a)
```



Gambar 19. Perhitungan Jumlah Botol (a) pada Serial Monitor (b) pada Aplikasi *Blynk*

Pada gambar 18 ditampilkan enam buah *Lihgt Emission Diode* (LED) sebagai indikator *plant* dan dua buah *value display* untuk menampilkan jumlah botol putih dan hitam. Gambar 19 menunjukkan perhitungan botol pada aplikasi *Blynk*.

Evaluasi Kinerja Sistem Setelah implementasi dan pengujian, dilakukan evaluasi terhadap kinerja sistem kontrol *Container Station* berbasis IoT yang telah dikembangkan. Evaluasi meliputi aspek-aspek berikut:

1. Efisiensi Operasional: Sistem ini terbukti dapat meningkatkan efisiensi operasional dengan meminimalkan waktu tunggu, mengurangi kesalahan manusia, dan mengoptimalkan alokasi sumber daya di *Container Station* (Nguyen et al., 2018; Sari et al., 2021).
2. Akurasi Data: Dengan adanya sensor dan sistem manajemen data yang terintegrasi, sistem ini mampu memberikan informasi yang akurat dan *real-time* mengenai lokasi, kondisi, dan pergerakan kontainer (Bragado et al., 2020; Zohri et al., 2019).
3. Visibilitas Pergerakan Kontainer: Antarmuka pengguna yang intuitif memungkinkan pemantauan *real-time* terhadap pergerakan kontainer, sehingga memudahkan pengambilan keputusan dan perencanaan operasional (Cao et al., 2021; Prasetyo et al., 2021).
4. Keamanan: Modul keamanan yang diintegrasikan ke dalam sistem terbukti efektif dalam mencegah kehilangan atau pencurian kontainer, serta memastikan keamanan data dan akses sistem (Cao et al., 2021; Mejjaoui & Babiceanu, 2018).
5. Dampak Keseluruhan: Secara keseluruhan, implementasi sistem kontrol *Container Station* berbasis IoT ini terbukti dapat meningkatkan produktivitas dan efisiensi operasional di *Container Station*, serta memberikan nilai tambah bagi perusahaan logistik dan pelanggan (Nguyen et al., 2018; Zhong et al., 2017).

Meskipun hasil evaluasi menunjukkan kinerja yang positif, terdapat beberapa tantangan dan peluang untuk pengembangan lebih lanjut. Tantangan yang dihadapi meliputi masalah integrasi dengan sistem lain dalam rantai pasokan, pengelolaan data yang semakin besar, serta kebutuhan pelatihan dan penyesuaian proses bagi pengguna (Bragado et al., 2020; Cao et al., 2021). Sementara itu, peluang untuk pengembangan lebih lanjut mencakup penerapan teknologi baru seperti *Machine Learning* atau *Blockchain*, serta kemungkinan untuk mengembangkan sistem serupa untuk fasilitas logistik lainnya (Liang et al., 2020; Mejjaoui & Babiceanu, 2018).

KESIMPULAN

Penelitian ini telah berhasil merancang dan mengimplementasikan sebuah sistem kontrol *Container Station* berbasis *Internet of Things* (IoT) dengan menggunakan metode *Waterfall*. Sistem ini dikembangkan dengan tujuan utama untuk meningkatkan efisiensi operasional, meminimalkan kesalahan manusia, dan memberikan visibilitas *real-time* terhadap pergerakan serta status kontainer di *Container Station*.

Berdasarkan hasil pengujian, integrasi antara perangkat *Programmable Logic Controller* (PLC) dan jaringan *Internet of Things* (IoT) dapat sukses terwujud dengan menggunakan *NodeMCU* sebagai perantara keduanya. Penggunaan aplikasi *Blynk* sebagai platform untuk membuat aplikasi kontrol dan pemantauan *container station* dengan menggunakan empat *push button*, enam *Lihgt Emission Diode* (LED), dan dua buah *value display*. Aplikasi *Blynk* yang telah dibuat dapat melakukan kontrol dan monitoring status *container station*.

DAFTAR PUSTAKA

- Alvi, S. N., Siddiqui, S. T., & Shah, M. A. (2021). Internet of Things (IoT) enabled container tracking and monitoring system. *IEEE Access*, 9, 51401-51412. DOI: 10.1109/ACCESS.2021.3069463
- Ashari, R. A., & Abbas, B. S. (2020). Aplikasi sistem pemantauan container berbasis IoT dan web. *Jurnal Infotel*, 12(2), 40-46. DOI: 10.20895/infotel.v12i2.516
- Aziz, A. H., Mutiara, L., Nugraha, S. A., & Astuti, D. P. (2019). Design and implementation of container tracking system based on Internet of Things (IoT). *Proceedings of International*

- Seminar on Application for Technology of Information and Communication (iSemantic), 185-189. DOI: 10.1109/ISEMANTIC.2019.8884311
- Bragado, L., Abt, T., & Ansari, N. (2020). IoT-based cargo tracking and monitoring system for smart logistics. *IEEE Access*, 8, 184845-184858. DOI: 10.1109/ACCESS.2020.3029762
- Cao, Y., Xie, H., Liu, J., & Chen, H. (2021). An integrated framework for container terminal management based on blockchain and internet of things. *IEEE Internet of Things Journal*, 8(16), 12832-12846. DOI: 10.1109/JIOT.2021.3065350
- Defi, S., Pagiling, L., Nur, M. N. A., Larobu, F. E., Jalil, A., & Badia, B. A. (2024). Rancang bangun prototipe sistem kontrol automatic main failure (AMF) menggunakan SMS (Short Message Service) berbasis PLC (Programmable Logic Controller). *Jurnal Mekanova: Mekanikal, Inovasi dan Teknologi*, 10(1), 75-87. <https://doi.org/10.12345/jmekanova.v10i1.75-87>
- Efendi, R., Darwin, D., Badia, B. A., Tando, A., Herlina, H., & Padang, W. L. (2023). Rancang bangun termokopel data logger berbasis Arduino Mega 2560 skala laboratorium. *Machine: Jurnal Teknik Mesin*, 9(2), 15-19. <https://doi.org/10.1109/machine.v9i2.15-19>
- Farooq, M. U., Waseem, M., Khairi, A., & Mazhar, S. (2015). A review on Internet of Things (IoT). *International Journal of Computer Applications*, 113(1), 1-7. DOI: 10.5120/19787-1571
- Gunawan, I., Akbar, T., & Ilham, M. G. (2020). Prototipe Penerapan Internet Of Things (Iot) Pada Monitoring Level Air Tandon Menggunakan Nodemcu Esp8266 Dan Blynk. *Jurnal Informatika Dan Teknologi*, 3(1), 1-7.
- Kusnandar, D., Pratama, P. A., & Setyowibowo, H. (2021). Sistem pengawasan kontainer berbasis Internet of Things (IoT) menggunakan teknologi LoRa. *Jurnal Teknik ITS*, 10(2), A350-A355. DOI: 10.12962/j23373539.v10i2.63225
- Larman, C., & Basili, V. R. (2023). Iterative and incremental development: A brief history. *IEEE Computer*, 36(6), 47-56. DOI: 10.1109/MC.2003.1204375
- Liang, Z., Feng, Y., Zhang, X., Wu, T., & Chaovalitwongse, W. A. (2020). A two-stage heuristic approach for the integrated optimisation of container terminal operations. *International Journal of Production Research*, 58(18), 5541-5558. DOI: 10.1080/00207543.2019.1651457
- Mejjaoui, S., & Babiceanu, R. F. (2018). RFID-wireless sensor networks integration for real-time ubiquitous cargo tracking. *Procedia Manufacturing*, 17, 170-177. DOI: 10.1016/j.promfg.2018.10.029
- Muliawan, A. D., Mahmudy, W. F., & Fauzi, M. A. (2022). Perancangan dan implementasi sistem monitoring dan manajemen container berbasis IoT. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 6(5), 5301-5310. DOI: <http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/10817>
- Nguyen, W. C., Chiu, C. H., Lin, C. H., & Lam, H. S. (2018). A geographic information system for managing and monitoring container terminals. *Computers & Industrial Engineering*, 117, 86-102. DOI: 10.1016/j.cie.2018.01.014
- Prasetyo, S., Yustiana, I., & A. Fergina. (2023). Rancang Bangun Automatic Liquid Filling Machine Berbasis Internet of Things Menggunakan NodeMCU dan Telegram. *Jurnal Ilmiah Pendidikan Teknik Elektro*, 7(1). DOI: 10.22373/crc.v7i1.14290
- Ramadhan, R., Putrada, M. S., & Meidrina, E. (2018). Sistem pemantauan container berbasis Internet of Things (IoT). *JUITA: Jurnal Informatika*, 6(2), 85-91. DOI: 10.30595/juita.v6i2.3100

- Rahayu, A. E., Faisal, S., & Pratama, A. R. (2021). Penghitung Kayu Kaso Otomatis Menggunakan Sensor Jarak Berbasis Internet of Things. *Scientific Student Journal for Information, Technology and Science*, 2(2), 184-190.
- Santoso, N. B., Supriyanto, A., & Rizaldi, M. (2021). Pengembangan Sistem Pemantauan Keamanan Ruang Berbasis Internet of Things (IoT). *Jurnal Ilmiah Teknologi Informasi*, 19(1), 42-49. DOI: 10.33557/resti.v5i1.3343
- Sari, D. P., Suryani, E., & Sukanto, A. S. (2021). Aplikasi E-Commerce Penjualan Makanan dan Minuman pada CV XYZ. *Jurnal Teknologi Informasi dan Komunikasi*, 12(2), 131-140. DOI: 10.21108/jtik.v12i2.7783
- Zohri, A., Sumardi, S., & Kusuma, A. H. P. (2019). Sistem Pemantauan Dan Pengendalian Ruang Berbasis Internet of Things. *Indonesian Journal of Computing and Cybernetics Systems*, 13(2), 151-160. DOI: 10.22146/ijccs.42618.