

# Prototipe Sistem Pemanggil Pelayan di Restoran Menggunakan Metode Nirkabel dan ESP32 dengan Protokol MQTT

Riski David Kasyanto<sup>1</sup>, Fikra Titan Syifa<sup>\*2</sup>, dan Slamet Indriyanto<sup>3</sup>

<sup>1,2,3</sup>Universitas Telkom Kampus Purwokerto

Jalan D.I. Panjaitan nomor 128 Purwokerto Banyumas Jawa Tengah, Kode Pos 53147, Indonesia

2212101134@ittelmkom-pwt.ac.id<sup>1</sup>, fikras@telkomuniversity.ac.id<sup>\*2</sup>, slamet@telkomuniversity.ac.id<sup>3</sup>

**Abstrak**—Pada penelitian ini telah dirancang sistem pemanggil pelayan berbasis sistem ESP32 dengan komunikasi nirkabel menggunakan protokol MQTT. Sistem terdiri dari tiga tombol utama, yaitu Call, Bill, dan Reset, yang digunakan pelanggan untuk mengirimkan permintaan secara langsung ke pelayan. Aktivitas makan siang yang sibuk dan akhir pekan, restoran sering mengalami kendala dalam memberikan pelayanan yang cepat dan responsif, yang berdampak pada menurunnya kepuasan pelanggan. Cara kerja sistem pemanggil pelayan secara nirkabel berupa data dikirim dalam format JSON melalui MQTT broker di cloud dan ditampilkan secara real-time, aplikasi Blynk. Pengujian dilakukan pada berbagai jarak. Hasil menunjukkan rata-rata throughput sebesar 723,536 bit/s. Packet loss tercatat 54,85%. Hasil pengujian jitter adalah 180,86 ms. Penerapan sistem ini terbukti efektif dalam mempercepat proses layanan, mengurangi waktu tunggu pelanggan, serta meningkatkan efisiensi operasional. Dengan pendekatan Internet of Things (IoT), sistem ini mendukung transformasi digital pada sektor layanan restoran. ini

Kata kunci— ESP32, mqtt, nirkabel, blynk, restoran.

**Abstract**— This research aims to design and implement a waitress calling system based on ESP32 microcontroller using wireless communication via the MQTT protocol. The system consists of three main buttons Call, Bill, and Reset used by customers to send requests directly to waitstaff. Activity such as lunch break time and weekends, restaurants often face challenges in providing fast and responsive service, leading to decreased customer satisfaction. Data is transmitted in JSON format through a cloud-based MQTT broker and displayed in real-time, the Blynk application. Testing was conducted at various distances. The results showed an average throughput of 723.536 bit/s. Packet loss was recorded at 54.85%. Result test of jitter is about 180.86 ms. The implementation of this system has proven effective in speeding up service processes, reducing customer waiting time, and improving operational efficiency. Using an Internet of Things approach, this system supports digital transformation in the restaurant service sector.

**Keywords**— ESP32, mqtt, wireless, blynk, restaurant.

## I. PENDAHULUAN

Kondisi restoran yang ramai pengunjung dapat berpengaruh terhadap pelayanan yang padat antrian pada sistem order restoran tersebut (Kepakisan et al., 2023). Pihak pramusaji dan manajemen mengontrol sistem *first in first out* (FIFO) pada SOP (*standard operational procedure*). Hal ini tentunya akan menimbulkan ketidakpuasan tamu atau konsumen restoran dengan pelayanan yang diberikan, misalnya jika pelanggan harus berteriak atau melambaikan tangan ketika ingin memanggil pelayan (Mardewi et al., 2023)

Salah satu aspek yang dapat memengaruhi operasional di sebuah restoran yaitu tentang kecepatan dan ketepatan dalam proses pemesanan menu. Proses pemesanan makanan (menu) dalam sebuah restoran masih menggunakan sistem manual atau konvensional (pencatatan). Hal ini akan menimbulkan persoalan seperti kesalahan manusia, antrian, keterlambatan. Jika dipandang dari sudut pandang konsumen maka akan menghasilkan komunikasi yang kurang efektif antara keduanya (tati et al., 2023).

Penerapan sistem order pemanggil pelayan di restoran dapat dilakukan secara *on the spot* menggunakan perangkat yang telah disediakan di meja konsumen. Sistem ini menggunakan nirkabel tanpa memerlukan interaksi langsung dengan pelayan. Restoran yang memiliki konsep area yang luas, ruangan keluarga (family) atau yang bersifat *outdoor* dengan pemandangan alam yang indah memungkinkan pelayan untuk berjalan kaki dengan jarak yang jauh (Prasetyo et al., 2024).

Sistem pemanggil pelayan (*waitress calling system*) merupakan salah satu alat bantu yang digunakan untuk memfasilitasi komunikasi tersebut. Sistem ini memungkinkan pelanggan restoran memanggil pelayan tanpa harus meninggalkan meja makan atau tempat duduk. Loyalitas pelanggan memegang peran penting dalam meningkatkan layanan (Kuswara et al., 2025). Dengan menggunakan teknologi nirkabel (*wireless*) yang berkembang saat ini, sistem pemanggil pelayan telah banyak dikembangkan.

Pada penelitian ini bertujuan untuk mengembangkan sistem pemanggil pelayan menggunakan mikrokontroler ESP32 dan protokol komunikasi MQTT untuk memudahkan pelanggan

memanggil pelayan hanya dengan menekan tombol. Sinyal kemudian dikirim secara nirkabel ke perangkat penerima. Perangkat ini terintegrasi dengan Blynk sebagai sistem pemantauan tambahan berbasis IoT.

## II. TINJAUAN PUSTAKA

*Wireless sensor network* yang mana disingkat dengan WSN diartikan jaringan sensor nirkabel. Istilah lebih lanjut merupakan jaringan dari sensor-sensor kecil yang saling terhubung tanpa menggunakan kabel fisik. Sensor-sensor ini memiliki kemampuan untuk mendeteksi berbagai parameter lingkungan, seperti suhu, kelembaban, tekanan, cahaya, suara, gerakan (Husin & Rahayu, 2025).

Teori antrian atau *queuing theory* pada sistem pemanggil pelayan atau pramusaji yang menggunakan perangkat teknologi nirkabel, mikrokontroler serta menggunakan protokol komunikasi sebagai media transmisi data dapat dikirimkan dengan cepat. Data yang dikirim oleh pelanggan diterima oleh pelayan (Ilmiah et al., 2021).

Selain itu, sebuah *waitress calling system* lainnya yang diintegrasikan dengan Internet of Things (IoT) telah dikembangkan, memungkinkan pelayan untuk memonitor panggilan pelanggan melalui dashboard secara *real-time*. Hal ini bertujuan untuk mengumpulkan, mengirimkan, dan menganalisis data dari lingkungan sekitarnya secara *real-time* atau dalam interval waktu tertentu. Data yang dikumpulkan oleh sensor-sensor ini digunakan untuk tujuan pemantauan, pengendalian, pemrosesan informasi, dan pengambilan keputusan.

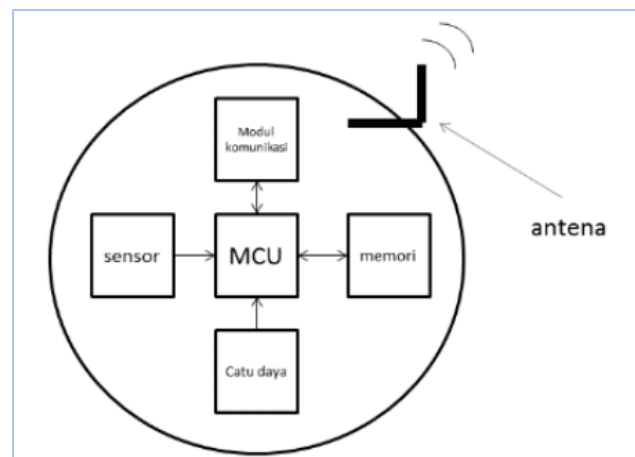
Berdasarkan keterbatasan tersebut, studi ini akan menggabungkan sistem pemanggil pelayan dengan metode *hardware and software integration* dengan pengembangan melalui perancangan desain perangkat keras dan pemanfaatan platform Node-RED dan Blynk Cloud. Selanjutnya, penelitian ini menggunakan mikrokontroler ESP32 sebagai perangkat utama dengan media transmisi Wi-Fi dan protokol MQTT yang dapat dimonitoring melalui dashboard Node-RED di PC pelayan/pramusaji.

## III. DASAR TEORI

Konsep dari suatu jaringan sensor nirkabel adalah berdasarkan fungsi penginderaan. *Node sensor* dalam WSN adalah unit dasar yang terdiri dari sensor fisik, komponen nirkabel, dan modul pengolahan data (Murdiyat & Gunanto, 2021). *Node sensor* bertugas untuk mendeteksi parameter lingkungan tertentu, mengumpulkan data dan mengirimkan data tersebut ke node koordinasi atau *sink node* dalam jaringan (Tech, n.d.). Sebuah *node sensor* umumnya terdiri dari beberapa komponen seperti sensor, unit prosesor, transmitter nirkabel, baterai atau sumber daya energi. *Node sensor* dalam WSN dapat beroperasi secara mandiri atau dalam kelompok (Wixted et al., 2016). Data yang dikumpulkan oleh node-node ini kemudian akan digunakan untuk analisis, pemantauan, pengambilan keputusan atau aplikasi lainnya yang sesuai dengan tujuan dari jaringan WSN tersebut.

Komponen penyusun node dalam WSN memiliki beberapa bagian yang berfungsi sebagai penghubung antara jaringan

sensor nirkabel (Subagiyo et al., 2023). Fungsi konektivitas antar jaringan melakukan transmisi untuk menghubungkan jaringan eksternal yang lebih luas seperti internet. Kemudian, secara teoritis konektivitas *node sensor* supaya terhubung melalui internet memiliki topologi (Wixted et al., 2016). Selanjutnya, jika sebuah perangkat *node sensor* dapat berkomunikasi dengan platform IoT maka memungkinkan untuk konfigurasi pemanfaatan jenis protokol komunikasinya. Pada penelitian ini menggunakan MQTT sebagai protokol komunikasi.

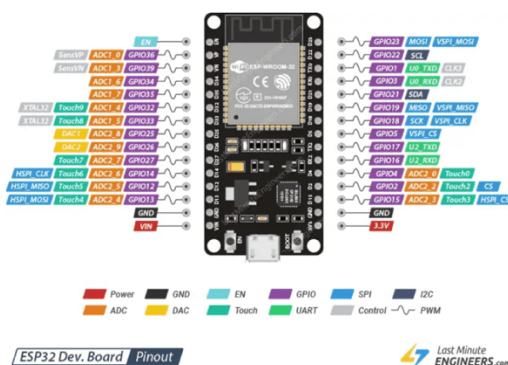


Gambar 1. Diagram Blok Konsep Node Sensor

Protokol komunikasi berbentuk lapisan-lapisan (*layering*) seperti TCP/IP dan OSI. *Layering* berfungsi untuk memecah kompleksitas pada suatu perangkat menjadi struktur yang lebih sederhana.

### A. ESP32 DEV Kit

ESP32 adalah *development board* dengan mikrokontroler yang merupakan penerus mikrokontroler ESP8266 yang terkenal dari Espressif. ESP32 adalah SoC berkemampuan Wi-Fi dan Bluetooth yang sangat kuat dengan jumlah GPIO yang lebih banyak. Selanjutnya, ESP32 ini adalah development board yang memiliki kelebihan berupa chip kombinasi konektivitas Wi-Fi dan Bluetooth 2,4GHz dan dirancang dengan daya rendah. Hal ini dirancang untuk mencapai kinerja daya dan RF terbaik (Systems, 2020).



Gambar 2. Pinout ESP32 DEV

Mikrokontroler ESP32 ini menggunakan prosesor dengan kecepatan 240 MHz, memori 520 KB, dan konektivitas jaringan nirkabel WiFi 802.11 b/g/n untuk dapat terhubung melalui protokol seperti MQTT. Fitur lain berupa ADC 12-bit, UART, I2C, SPI, PWM, hingga port GPIO. Integrasi *hardware* ini dapat digunakan untuk konsep *node sensor*.

## B. Topologi Jaringan

Topologi jaringan adalah didefinisikan sebagai suatu teknis, cara dan aturan di dalam merangkai dan menghubungkan berbagai komputer dan perangkat terhubung lainnya ke dalam sebuah jaringan computer sehingga membentuk sebuah hubungan yang bersifat geometris. Topologi ini bersifat sebuah rancangan atau desain, yang dapat diimplementasikan secara langsung melalui sejumlah perangkat keras penghubung pada jaringan komputer.

Topologi juga dapat diartikan sebagai layout atau arsitektur atau diagram jaringan komputer. Topologi merupakan suatu aturan/rules bagaimana menghubungkan komputer (node) secara fisik. Topologi berkaitan dengan cara komponen-komponen jaringan saling berkomunikasi melalui media transmisi data. Topologi jaringan menjelaskan bagaimana perangkat-perangkat tersebut terhubung satu sama lain untuk berkomunikasi dan bertukar data.

## C. Blynk

Blynk adalah platform yang bertujuan untuk kendali modul Arduino, Raspberry Pi, ESP8266, Wemos D1, dan modul sejenisnya melalui internet. Aplikasi Blynk merupakan platform untuk membuat antarmuka grafis untuk project yang akan diimplementasikan hanya dengan metode *drag and drop widget* sehingga sangat mudah digunakan. Platform ini sangat mudah digunakan untuk *monitoring and controlling* berbagai perangkat IoT karena pemrograman pada board yang tidak rumit dan konfigurasi pada *mobile app* yang mudah.

## D. Quality of Services

Quality of Service (QoS) merupakan metode pengukuran tentang seberapa baik jaringan dan merupakan suatu usaha untuk mendefinisikan karakteristik dan sifat dari satu servis. QoS digunakan untuk mengukur sekumpulan atribut kinerja yang telah dispesifikasikan dan diasosiasikan dengan suatu servis. Model monitoring QoS terdiri dari komponen monitoring application, QoS monitoring, monitor, dan monitored object.

*Monitoring application* merupakan sebuah antarmuka bagi administrator jaringan. Komponen ini berfungsi mengambil informasi lalu lintas paket data dari monitor, menganalisis, dan mengirimkan hasil analisis kepada pengguna. QoS monitoring menyediakan mekanisme monitoring QoS dengan mengambil informasi nilai-nilai parameter QoS dari lalu lintas paket data. Monitor melakukan pengukuran aliran paket data secara waktu nyata dan melaporkan hasilnya kepada *monitoring application*.

## E. Protokol MQTT (Message Queuing Telemetry Transport)

*Message Queuing Telemetry Transport* (MQTT) adalah protokol M2M (*machine to machine*). MQTT didesain sebagai

protokol transport yang sangat ringan dengan konsep *publish/subscribe*. Sangat bermanfaat untuk membangun aplikasi yang digunakan pada lokasi kurang terjangkau jaringan internet dan harga bandwidth yang mahal. MQTT juga ideal digunakan untuk *mobile application* karena ukuran datanya yang kecil dan penggunaan power yang rendah (Susanto et al., 2018).

MQTT dibentuk untuk menjadi *low-overhead protocol* dengan mempertimbangkan kebutuhan bandwidth dan keterbatasan dari kemampuan komputasi sebuah CPU. MQTT didesain dengan kemampuan untuk berjalan pada *embedded environment* dimana MQTT efektif untuk menghadirkan komunikasi yang *real-time*. MQTT sangat prospektif seiring dengan pertumbuhan pasar *internet of things* yang pesat (Mumtaz et al., 2023).

Prinsip kerja protokol MQTT yaitu menggunakan konsep *publish/subscribe* yang sangat berbeda dengan konsep protokol yang lain seperti HTTP yang menggunakan paradigma *request/response*. Titik pusat komunikasi adalah MQTT *broker* yang bertanggung jawab untuk mengirim semua pesan diantara pengirim dan penerima (Kurnianto et al., 2022).

Setiap client yang mengirim pesan ke *broker* harus menyisipkan informasi nama *topic* yang dituju (Informasi et al., 2022). *Topic* adalah informasi *routing* untuk *broker* yang bertugas mengarahkan pesan yang masuk kedalam *broker*. Client yang ingin menerima pesan harus melakukan *subscribe* ke *topic* yang dituju dan selanjutnya *broker* akan mengirimkan semua pesan yang diarahkan pada topik tersebut. Oleh karena itu client tidak perlu saling mengenal dan dapat berkomunikasi melalui *topic* ini.

Perbedaan mendasar dengan protokol HTTP adalah client tidak perlu menarik informasi yang dibutuhkannya tetapi *broker* akan mengirimkan informasi ke client jika ada pesan baru yang masuk ke sebuah *topic*. Jika koneksi mengalami gangguan maka MQTT *broker* dapat menyimpan semua pesan yang masuk dan mengirimkannya kembali ke client ketika koneksi kembali normal (Monitoring et al., 2024).

MQTT Client dapat bertindak sebagai *publisher* ataupun *subscriber*. MQTT Client dapat berupa peralatan mikrokontroler maupun aplikasi komputer yang ditanamkan library untuk dapat melakukan koneksi ke MQTT *broker*. Implementasi protokol MQTT pada client mudah untuk ditransmisikan pada perangkat (Putri & Putri, n.d.). Hal ini merupakan salah satu alasan mengapa MQTT ideal dan cocok untuk *small devices*. MQTT *broker* adalah *central device* yang berfungsi untuk memproses komunikasi antara MQTT client dan mendistribusikan pesan pada *topic* yang dituju. MQTT *broker* dapat menangani ribuan *devices* dalam satu waktu.

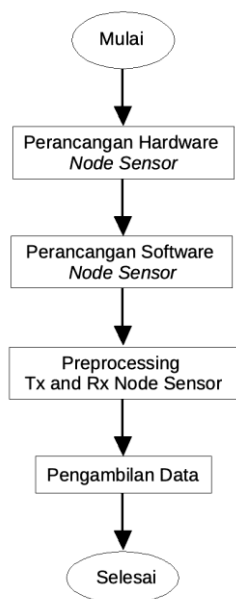
Arsitektur MQTT terdiri dari tiga komponen yaitu *publisher*, *broker*, dan *subscriber*. Sebuah perangkat atau *device* yang terhubung pada suatu *topic* harus diregistrasikan pada *topic* tersebut sebagai *subscriber* sehingga ketika *publisher* telah *publish status* informasi pada *topic* tersebut maka *broker* akan meneruskan informasi ke *subscriber*. Otorisasi *subscriber* dan *publisher* akan dilakukan verifikasi oleh *broker* untuk memastikan keamanan terjaga sehingga bisa



dipastikan bahwa *publisher* dan *subscriber* telah diregistrasikan pada broker (Informasi et al., 2022).

#### IV. METODE PENELITIAN

Alur penelitian dimulai dengan perancangan *hardware* atau perangkat keras *node sensor* seperti modul *transmitter* dan *receiver*. Tahap berikutnya adalah perancangan desain *software* pada platform Blynk termasuk sistem protokol MQTT. Setelah perangkat terhubung pada jaringan komunikasi antar perangkat maka tahapan berikutnya adalah mempersiapkan node sensor tersebut atau istilahnya adalah *preprocessing Tx and Rx node sensor* untuk pembacaan data dan pengambilan data. Gambar 3 menjelaskan alur penelitian.



Gambar 3. Alur Penelitian

Berikutnya, perancangan node sensor ini terdiri dari tiga komponen utama yaitu input, proses, dan output. Pada bagian input, pengguna menekan tombol fisik yang tersedia di masing-masing meja untuk mengirim permintaan seperti *call*, *bill*, atau *reset*. ESP32 akan membaca input ini dan mengubahnya menjadi data JSON berisi informasi seperti ID meja, jenis permintaan. Data ini dikirim melalui protokol MQTT melalui *broker* yang ditempatkan pada *cloud*. Gambar 4 menjelaskan tentang input, proses, dan output. Pada bagian proses, ESP32 receiver yang *subscribe topic* tertentu akan menerima pesan tersebut dan meneruskannya ke beberapa perangkat.



Gambar 4. Diagram Alur Sistem

Perancangan desain merupakan tahap penting dalam pengembangan *waitress calling system* agar dapat divisualisasikan secara nyata sebelum diimplementasikan. Pada tahap ini, dilakukan pembuatan bentuk fisik perangkat

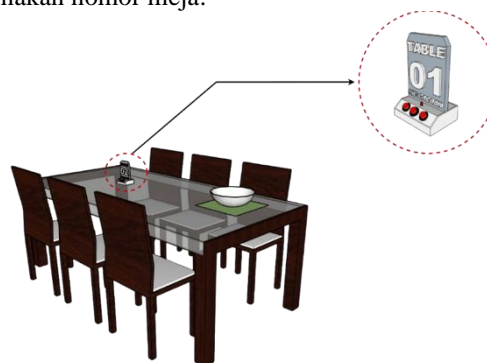
pengirim dan penerima yang telah disesuaikan dengan kebutuhan fungsional serta kondisi lingkungan operasional di restoran.

#### A. Penerapan Node Transmitter dalam Rangkaian



Gambar 5. Perancangan Desain Prototipe Transmitter (Tx)

Pada desain perangkat yang dirancang, digunakan untuk membantu pelanggan dalam memanggil pelayan menggunakan komunikasi nirkabel. Desain pada gambar 6 dirancang dengan menggunakan nomor meja.



Gambar 6. Penempatan Desain dari Transmitter (Tx)



Gambar 7. Perancangan Desain Prototipe Transmitter (Tx)

Berikut pada gambar 7 merupakan gambar desain wujud *node transmitter* yang mana terdapat 3 (tiga) buah *push button*.

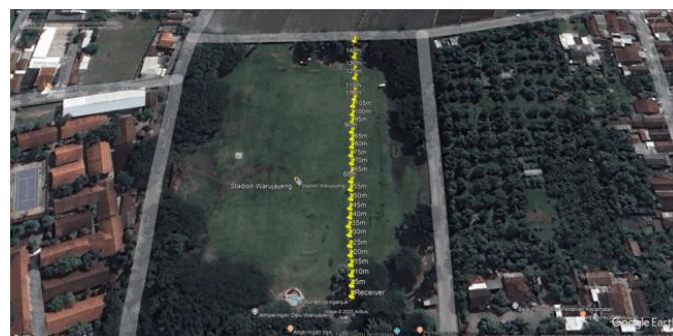
## B. Penerapan Node Receiver dalam Rangkaian



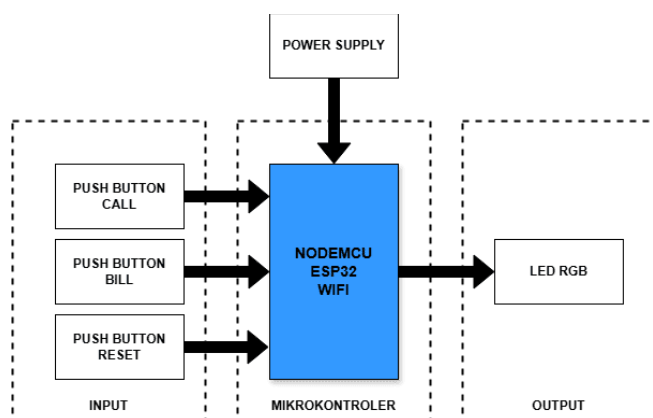
Gambar 8. Perancangan Desain Prototipe Receiver (Rx)

## V. HASIL PENGUJIAN SISTEM

Pengujian *waitress calling system* pada kondisi *line of sight* (LOS) dilakukan dengan variasi jarak mulai dari 5 meter hingga 150 meter guna mengukur performa dan batas jangkauan komunikasi nirkabel antara node pengirim (ESP32 Transmitter) dan node penerima (ESP32 Receiver) yang berkomunikasi melalui broker MQTT berbasis *cloud*. Tujuan utama dari pengujian ini adalah untuk menilai tingkat kestabilan protokol MQTT. Gambar 11 merupakan denah lokasi pengujian di areal terbuka atau *outdoor*.

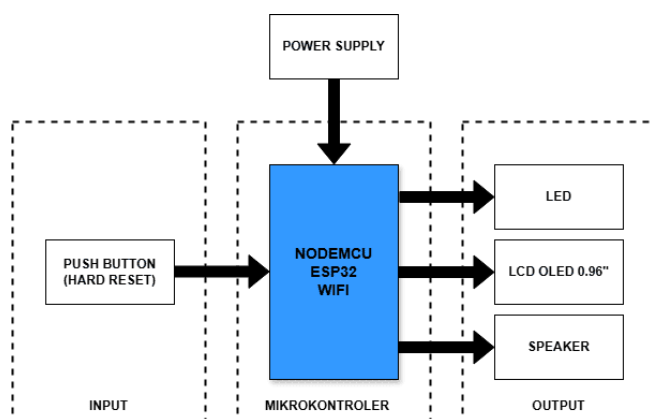


Gambar 11. Denah lokasi Pengujian *Line of Sight* (LOS)



Gambar 9. Blok Diagram Node Transmitter (Tx)

Pada perangkat transmitter menggunakan 3 (tiga) *push button*, yaitu *push button call*, *push button bill*, dan *push button reset*.



Gambar 10. Blok Diagram Node Receiver (Rx)

Hasil pengujian berikut ini merupakan kinerja dari sistem yang dijabarkan ke dalam bentuk tabel 1 dengan keterangan jarak dan tingkat keberhasilan pengiriman pada kondisi *line of sight* (LOS). Parameter jarak dimulai dari titik jarak 5 meter hingga titik terjauh 150 meter tanpa adanya penghalang dengan lokasi berada di luar ruangan atau *outdoor*.

### A. Hasil Pengujian Jarak Kondisi Line of Sight

Berdasarkan tabel 1 maka sistem menunjukkan kinerja optimal melakukan transmisi data hingga 90 meter. Pada rentang tersebut komunikasi antara *node transmitter* dan *node receiver* berjalan stabil. Hasil pengujian jarak menunjukkan terjadi peningkatan *latency* atau *time-out* ketika modul *transmitter* dan *receiver* sudah lebih dari 90 meter.

TABEL I. HASIL PENGUJIAN TX DAN RX PARAMETER JARAK

Nomor	Jarak (meter)	Keterangan
1	5	Sukses Terkirim
2	10	Sukses Terkirim
3	20	Sukses Terkirim
4	30	Sukses Terkirim
5	40	Sukses Terkirim
6	50	Sukses Terkirim
7	60	Sukses Terkirim
8	70	Sukses Terkirim
9	80	Sukses Terkirim
10	90	Gagal

11	100	Gagal
12	110	Gagal
13	120	Gagal
14	130	Gagal
15	140	Gagal
16	150	Gagal

Selanjutnya, dalam pengujian ini performa sistem komunikasi nirkabel juga bergantung pada kondisi dan jalur transmisi. Evaluasi diperlukan untuk menganalisis hambatan fisik antara *node transmitter* dan *node receiver*.

Prosedur pengujian dilakukan dengan menempatkan ESP32 Tx (*transmitter*) di berbagai titik dengan jarak yang berbeda dari sumber jaringan, yaitu hotspot smartphone, yang juga menjadi titik pusat koneksi Wi-Fi untuk keseluruhan sistem. Sementara itu, ESP32 Receiver diletakkan di lokasi tetap yang berada dekat dengan router Wi-Fi untuk memastikan stabilitas koneksi. Lokasi pengujian dilakukan di area terbuka Stadion Warujayeng untuk menciptakan kondisi LOS yang ideal, yaitu tanpa adanya penghalang fisik antara *transmitter* dan *receiver* sehingga pengaruh gangguan dapat diminimalkan.



Gambar 12. Denah lokasi Pengujian *Line of Sight* (LOS)

#### B. Hasil Pengujian Jarak Kondisi Non-Line of Sight

Berikutnya adalah pengujian jarak *non-line of sight* dengan jangkauan variasi hingga 100 meter. Tujuannya adalah untuk mengetahui batas performa sistem komunikasi nirkabel antara *node transmitter* dengan *node receiver*.

TABEL II. HASIL PENGUJIAN JARAK KONDISI NON-LINE OF SIGHT

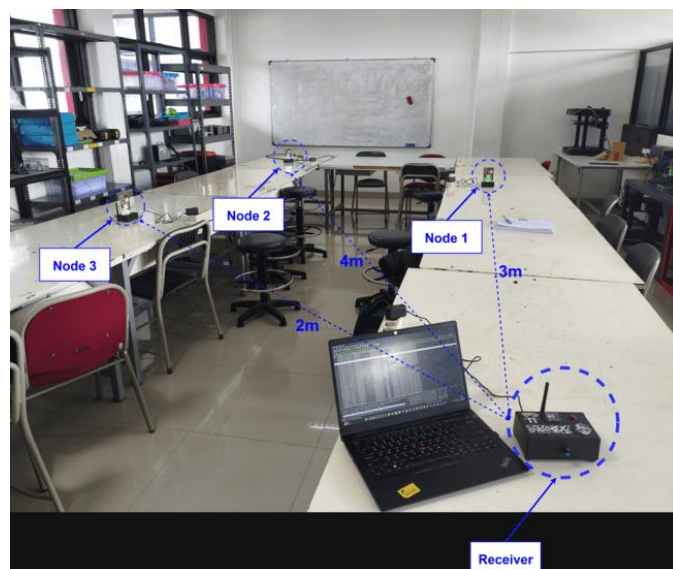
Nomor	Jarak (meter)	Keterangan
1	5	Sukses Terkirim
2	10	Sukses Terkirim
3	20	Sukses Terkirim
4	30	Sukses Terkirim
5	40	Sukses Terkirim
6	50	Sukses Terkirim
7	60	Gagal
8	70	Gagal
9	80	Gagal
10	90	Gagal
11	100	Gagal

#### C. Hasil Pengujian Jarak Dalam Ruangan (*indoor*)

Pada bagian ini, dijelaskan tata letak perangkat di dalam ruangan. Secara performa, semakin jauh jarak maka nilai *delay* akan semakin tinggi dan kegagalan dalam pengiriman data. Performa pengiriman data masih dapat dilakukan meski masih terdapat *delay*. Masing-masing data dapat terkirim dan memiliki jalur sendiri. Tabel 3 di bawah ini merupakan pengujian sistem transmisi jaringan terhubung internet cloud Blynk dengan mode aktual data (selisih *delay*). Parameter *delay* untuk pengujian yang dilakukan yaitu pengiriman pesan oleh *transmitter* ESP32 dan diterima oleh *receiver* ESP32. Gambar

TABEL III. TIMESTAMP HASIL PENGUJIAN DELAY INDOOR

Nomor	Timestamp (send)	Timestamp (receive)	Selisih Delay (ms)
1	08:28:06	08:28:06	0
2	08:28:11	08:28:12	1000
3	08:28:16	08:28:16	0
4	08:28:21	08:28:22	1000
5	08:28:26	08:28:26	0
6	08:28:31	08:28:31	0
7	08:28:36	08:28:37	1000
8	08:28:41	08:28:41	0
9	08:28:46	08:28:46	0
10	08:28:51	08:28:51	0



Gambar 13. Pengujian ESP32 Transmitter dan ESP32 Receiver

#### D. Hasil Pengujian *Jitter*

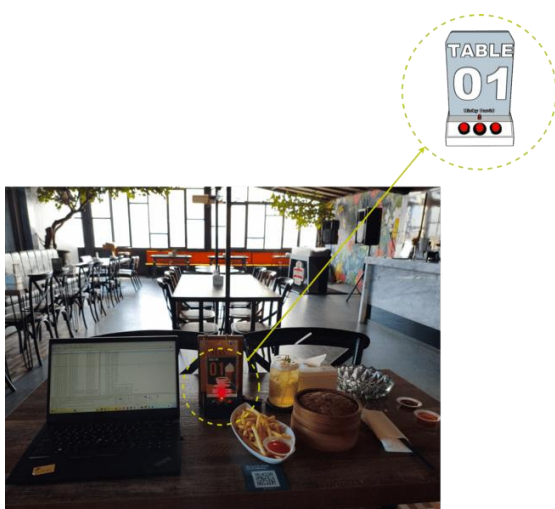
Pengujian *jitter* bertujuan mengukur variasi waktu tempuh antar paket data selama transmisi. Berdasarkan analisis menggunakan *Wireshark* pada berbagai jarak, diperoleh nilai *jitter* seperti tercantum dalam Tabel.



TABEL IV. HASIL PENGUJIAN JITTER

Nomor	Jarak (meter)	Variasi Delay (ms)	Paket yang diterima	Jitter (ms)
1	5	270	99	2,73
2	5	424	98	4,33
3	10	350	99	3,54
4	10	320	99	3,23
5	15	406	95	4,27
6	15	310	99	3,13
7	20	270	99	2,73
8	20	360	99	3,64
9	25	1925	66	29,17
10	25	780	90	8,67
11	30	190	99	1,92
12	30	663	94	7,05
5	40	695	94	7,39
6	50	793	91	8,71
7	60	0	-1	0
8	70	0	-1	0
9	80	0	-1	0
10	90	0	-1	0
Rerata Jitter (ms)				180,86

Pengujian dilakukan sampling data hingga 10 kali dan data dikirim pada berbagai jarak mulai dari 5 m hingga 90 m. Hasil pengujian menunjukkan bahwa pada jarak 5 m hingga 20 m masih dalam batas wajar. Sedangkan pada jarak 60 m hingga 90 m jitter tidak maksimal dikarenakan sistem komunikasi transmisi data mengalami kondisi menurun drastis dikarenakan bertambah jaraknya.



Gambar 14. Penempatan Prototipe Node Transmitter

Pada gambar 14 menunjukkan penempatan perangkat node transmitter pada meja pelanggan. Setiap node transmitter tersebut terpasang push button atau tombol serta LED indikator.



Gambar 15. Penempatan Prototipe Node Receiver

Berikutnya adalah gambar 15 yang mana merupakan bagian *node receiver* yang ditempatkan pada areal meja layanan atau pramusaji. *Node receiver* ini terdapat lampu indikator dan antena gambar menunjukkan penempatan perangkat *node transmitter* pada meja pelanggan

#### E. Hasil Pengujian *Packet Loss*

Pengujian *Packet Loss* bertujuan untuk mengidentifikasi prosentase paket data yang gagal sampai tujuan. Berdasarkan hasil tabel 6 maka pengujian dilakukan pada berbagai jarak hingga dilakukan pengujian data 20 kali.

TABEL V. HASIL PENGUJIAN PACKET LOSS

Pengujian ke-n	Jarak (meter)	Paket Data Dikirim	Paket Data Diterima	Packet Loss (%)
1	5	100	100	0
2	5	100	99	1
3	10	100	100	0
4	10	100	100	0
5	15	100	96	4
6	15	100	100	0
7	20	100	100	0
8	20	100	100	0
9	25	100	67	33
10	25	100	91	9
11	30	100	100	0
12	30	100	95	5
13	35	100	90	10
14	35	100	100	0
15	40	100	95	5
16	40	100	93	7

17	45	100	38	62
18	45	100	43	57
19	50	100	92	8
20	50	100	84	16
Rerata Packet Loss (%)				54,85

Selanjutnya, berdasarkan tabel 6 tersebut menunjukkan bahwa pada jarak pendek 5 m hingga 30 m maka nilai packet loss relatif rendah. Namun, pada jarak diatas 35 m hingga 50 m data menunjukkan terjadi peningkatan packet loss yang cukup signifikan. Kondisi ini merupakan melemahnya sinyal dikarenakan jarak yang semakin jauh. Hal ini secara langsung mengindikasikan bahwa sinyal terputus.

#### F. Hasil Pengujian Throughput

Pengujian *throughput* bertujuan untuk mengukur kecepatan *actual* transfer data dalam jaringan. Tabel 7 di bawah ini menunjukkan bahwa pada parameter jarak diperoleh nilai *throughput* yang bervariasi seiring dengan semakin meningkatnya jarak. Pengujian data dilakukan sebanyak 22 kali dengan jarak 5 m hingga 55 m. Hal ini fokus pada performa sistem yang berperan sebagai indikator transmisi data.

TABEL VI. HASIL PENGUJIAN THROUGHPUT

No	Jarak (m)	Jumlah Data Dikirim (bytes)	Throughput (bytes/s)	Throughput (bit/s)
1	5	19192	191,92	1535,36
2	5	19138	191,38	1531,04
3	10	19192	191,92	1535,36
4	10	19192	191,92	1535,36
5	15	18976	189,76	1518,08
6	15	191,92	191,92	1535,36
7	20	19192	191,92	1535,36
8	20	19192	191,92	1535,36
9	25	17410	174,1	1392,8
10	25	18706	187,06	1496,48
11	30	19192	191,92	1535,36
12	30	18922	189,22	1513,75
13	35	17686	176,86	1414,88
14	35	19192	191,92	1535,36
15	40	18922	189,22	1513,76
16	40	18814	188,14	1505,12
17	45	9771	97,71	781,68
18	45	8247	82,47	659,76
19	50	18483	82,47	659,76
20	50	18051	180,51	144,08
21	55	4713	47,13	377,04
22	55	393	3,93	31,44
Rerata Throughput (bit/s)				723,536

Berikutnya, berdasarkan tabel 7 tersebut menunjukkan bahwa pada rentang jarak 5 m hingga 55 meter protokol MQTT mampu mempertahankan *throughput* yang stabil dengan kisaran 1400-1500 bit/s dengan nilai tertinggi mencapai 1535,36 bit/s. Hal ini menunjukkan bahwa dalam jarak pendek dan menengah masih memiliki konektivitas jaringan yang mampu mengirimkan data secara optimal. Selanjutnya, rerata *throughput* seluruh pengujian adalah 723,536 bit/s. Hal ini menunjukkan bahwa pentingnya desain perancangan jaringan dengan mempertimbangkan jarak efektif untuk mendapatkan hasil yang maksimal.

#### VI. KESIMPULAN

Hasil pengujian transmisi dengan *node transmitter* (Tx) dan *node receiver* (Rx) menunjukkan bahwa diatas rentang 90 meter kualitas jaringan akan terputus karena semakin jauhnya jarak. Kemudian pada pengujian *jitter* menunjukkan bahwa nilai rata-rata didapatkan angka 180,86 bit/s. Pengujian *packet loss* berhasil mendapatkan nilai rerata 54,85 %. Sedangkan untuk pengujian *throughput* mendapatkan nilai rerata sistem pada angka 723,536 bit/s.

#### REFERENSI

- Husin, I., & Rahayu, R. R. (2025). *PROTOTYPE SISTEM PENGAMBILAN PESANAN BERBASIS WIRELESS SENSOR NETWORK ( WSN ) TERINTEGRASI DENGAN APLIKASI CAFE*. 2, 164–171.
- Ilmiah, J., Mesin, T., & Komputer, E. D. A. N. (2021). *RANCANG BANGUN PROTOTYPE DISPLAY ANTRIAN PELANGGAN DENGAN*. 1(3).
- Informasi, F. T., Budioko, T., & Digital, U. T. (2022). *Node Sensor Pada Sistem Monitoring Tinggi Permukaan Air Sungai Berbasis FreeRTOS dan*. 1(1), 36–46.
- Kepakisan, K., Koerniawaty, F., & Kuntariati, U. (2023). Pengaruh Pelayanan Pramusaji dan Kualitas Produk Terhadap Kepuasan Konsumen. *Jurnal Ilmiah Pariwisata Dan Bisnis*, 2, 2007–2020. <https://doi.org/10.22334/paris.v2i9.539>
- Kurnianto, A., Irawan, J. D., & Industri, F. T. (2022). *PENERAPAN IOT ( INTERNET OF THINGS ) UNTUK CONTROLLING LAMPU MENGGUNAKAN PROTOKOL MQTT BERBASIS WEB*. 6(2).
- Kuswara, S. R., Quality, U., Pelayanan, K., & Loyalitas, D. (2025). *PENGARUH SELF SERVICE TECHNOLOGY DAN KUALITAS PELAYANAN TERHADAP LOYALITAS PELANGGAN MIE GACOAN ( PADA MAHASISWA PROGRAM STUDI MANAJEMEN UNIVERSITAS QUALITY ) THE EFFECT OF SELF SERVICE TECHNOLOGY AND SERVICE QUALITY ON CUSTOMER LOYALTY OF MIE GACOAN ( ON STUDENTS OF MANAGEMENT STUDY PROGRAM OF QUALITY UNIVERSITY )*. 4, 1–15.
- Mardewi, M., Iskandar, I., Sofyan, S., Wungo, S., & Aziz, F. (2023). Sistem Monitoring Status Meja Pada Restoran Berbasis Internet of Things (IOT). *Journal of System and Computer Engineering (JSCE)*, 4, 136–144. <https://doi.org/10.61628/jsce.v4i2.816>



- Monitoring, S., Berbasis, S., Dan, N., Menggunakan, W., & Nodemcu, M. (2024). *No Title*. 2, 238–243.
- Mumtaz, R., Akbar, R., Arif, T. Y., & Irhamisyah, M. (2023). *Analisis Performansi Protokol MQTT Pada Sistem Pemantauan Kualitas Udara Ruangan Berbasis IoT*. 8(3), 102–109.
- Murdiyati, P., & Gunanto, L. H. (2021). *Rancang Bangun Sensor Node Untuk Sistem Monitoring Energi Listrik Nirkabel Pada Gedung Dalam Kampus Politeknik Negeri Samarinda*. 2(1), 1–9.
- Pasinella, B. (2016). Resilient Energy Systems: Resilient Energy Design. In *Second Nature*.
- Prasetyo, L., Aryanta, S., Saputra, M., & Rahman, R. A. (2024). *Digital Transformation : Implementation of Wireless Calling System Case Study at Mie Gacoan Veteran Semarang from a Blue Ocean Strategy Perspective*. 535–540.
- Putri, P., & Putri, N. R. (n.d.). *MENGGUNAKAN MQTT PROTOCOL MELALUI*.
- Subagiyo, H., Rhamadani, R., Teknologi, P., Sistem, R., & Riau, P. C. (2023). *Penghematan Daya Pada Sensor Node Sistem Monitoring Kualitas Udara*. 13(2), 217–228.
- Susanto, B. M., Setiyawan, E., Atmadji, J., & Brenkman, W. L. (2018). *IMPLEMENTASI MQTT PROTOCOL PADA SMART HOME SECURITY BERBASIS WEB*. 4, 201–205.
- Systems, E. (2020). ESP8266EX Datasheet Version 6.0. *Espressif*.  
<https://www.espressif.com/en/subscribe>.
- Szabo, T. L. (2014). Diagnostic Ultrasound Imaging: Inside Out. In *Diagnostic Ultrasound Imaging: Inside Out: Second Edition*. Elsevier. <https://doi.org/10.1016/C2011-0-07261-7>
- tati, F., Kasoep, W., & Novani, N. (2023). Prototipe Pendeteksi Keberadaan Pelanggan Pada Meja Restoran Berbasis Mikrokontroler dengan Notifikasi Telegram: Indonesian. *CHIPSET*, 4, 88–92.  
<https://doi.org/10.25077/chipset.4.01.88-92.2023>
- Tech, K. (n.d.). *DESAIN NODE SENSOR UNTUK MONITORING LAHAN PERSAWAHAN BERBASIS MIKROKONTROLLER ATMEGA328P*. 99–106.
- Wixted, A. J., Kinnaird, P., Larijani, H., Tait, A., Ahmadinia, A., & Strachan, N. (2016). Evaluation of LoRa and LoRaWAN for wireless sensor networks. *2016 IEEE SENSORS*, 1–3.  
<https://doi.org/10.1109/ICSENS.2016.7808712>
- Wunderling, T., Golla, B., Poudel, P., Arens, C., Friebe, M., & Hansen, C. (2017). *Comparison of thyroid segmentation techniques for 3D ultrasound* (M. A. Styner & E. D. Angelini (eds.); p. 1013317).  
<https://doi.org/10.1117/12.2254234>