

## IDENTIFIKASI RADIUS SENSOR NODE -GATEWAY PADA SISTEM IOT BERBASIS PROTOKOL MQTT

### *THE IDENTIFICATION OF SENSOR NODES-GATEWAY RADIUS IN IOT SYSTEM BASED ON MQTT PROTOCOL*

Stephanie I Pella<sup>1)</sup>, Hendro FJ Lami<sup>2)</sup>

<sup>1, 2)</sup>Prodi Teknik Elektro Universitas Nusa Cendana

Jl. A. Sucipto Penfui Kupang

<sup>1)</sup>e-mail: [s.i.pella@gmail.com](mailto:s.i.pella@gmail.com)

#### ABSTRAK

Radius antara sensor node dan gateway dalam suatu sistem komunikasi WSN ditentukan melalui parameter received signal strength indicator (RSSI). Fluktuasi nilai RSSI terjadi karena adanya perubahan posisi sensor node terhadap gateway. Penelitian ini akan menggunakan RSSI sebagai parameter dalam mendisain suatu sistem identifikasi radius esp8266 yang berfungsi sebagai sensor node. Sistem dibangun menggunakan Node-RED dengan protokol MQTT sebagai protokol komunikasi antar sensor node dan gateway pada frekuensi 2.4GHz. Pengukuran jarak referensi LOS antara sensor node dan gateway pada jarak 1m memiliki nilai rssi sebesar -35dBm. Nilai tersebut menjadi inputan fungsi pada Node-Red yang berjalan pada gateway. Gateway memiliki sebuah database untuk melakukan proses estimasi radius sensor node. Hasil pengujian menunjukkan bahwa nilai error rata-rata yang terjadi pada saat identifikasi radius memiliki nilai sebesar 1.85%.

**Kata Kunci:** RSSI, WSN, MQTT, LOS

#### ABSTRACT

*The radius between the sensor node and the gateway in a WSN communication system is determined through the received signal strength indicator (RSSI) parameter. The fluctuation of the RSSI value occurs due to the distance between sensor nodes and gateway. This study will use RSSI as a parameter in designing an esp8266 radius identification system that functions as a sensor node. The system is built using Node-RED with the MQTT protocol as a communication protocol between sensor nodes and gateways at a frequency of 2.4GHz. Measurement of the LOS reference distance between the sensor node and the gateway at a distance of 1m has an RSSI value of -35dBm. This value becomes the input function on Node-Red running on the gateway. Gateway has a database to perform the process of estimating the radius of the sensor node. The test results show that the average of the radius identification has a value of 1.85%.*

**Keywords:** RSSI, WSN, MQTT, LOS

#### **PENDAHULUAN** (Arial 11, Bold, spasi 1,5, spacing before 12 pt, after 6 pt)

Received Signal Strength Indicator (RSSI) merupakan parameter dalam sistem wireless sensor network (wsn) yang menentukan kualitas sinyal penerimaan terhadap perubahan posisi antara suatu sensor node dan gateway. Lingkungan propagasi line of sight (LoS) maupun no line of sight (NLoS) ikut mempengaruhi fluktuasi nilai rssi tersebut [Chuku, et al, 2021; Johndale, et al, 2020]. Selain itu terdapat beberapa faktor yang ikut mempengaruhi yaitu daya pancar gateway dan sensor node, gain antenna gateway maupun sensor node, serta frekuensi kerja dari sistem tersebut.

Perangkat mikrokontroler saat ini menyediakan fasilitas-fasilitas untuk mendukung pengembangan integrasi wsn dan Internet of Things(IoT)[Kurniawan, 2019]. Salah satu perangkat yang menyediakan fasilitas tersebut adalah esp8266. Esp8266 memiliki kemampuan terkoneksi dengan jaringan nirkabel pada frekuensi 2.4GHz[ESP8266EX Datasheet , 2021]. Perangkat ini memiliki antena internal dengan tiga kondisi daya pancar yaitu -20dBm untuk mode 802.11b, -17dBm untuk mode 802.11g dan -14dBm untuk mode 802.11n. Melalui tiga mode komunikasi tersebut esp8266 mampu berfungsi sebagai pemancar maupun penerima dan dapat berkomunikasi pada 14 channel frekuensi untuk mode akses point.

Proses penentuan mode kerja dari esp8266 sebagai pemancar ataupun penerima dapat menggunakan bahasa pemrograman tertentu misalnya bahasa c, c++, java, maupun python[Vasilev, D. 2020]. Selain bahasa-bahasa pemrograman tersebut terdapat tool programming untuk memudahkan komunikasi antar sensor node. Node-Red merupakan flow-based platform yang memudahkan pengembang aplikasi IoT dimana melalui tool tersebut sensor node dapat berkomunikasi menggunakan beberapa protokol misalnya http dan mqtt[Node-Red, 2021; sicari, et all, 2019].

Berdasarkan pertimbangan kemampuan esp8266 yang memiliki modul komunikasi nirkabel maka penelitian ini akan memodelkan penentuan radius sensor node melalui level rssi menggunakan protokol komunikasi mqtt menggunakan node-red.

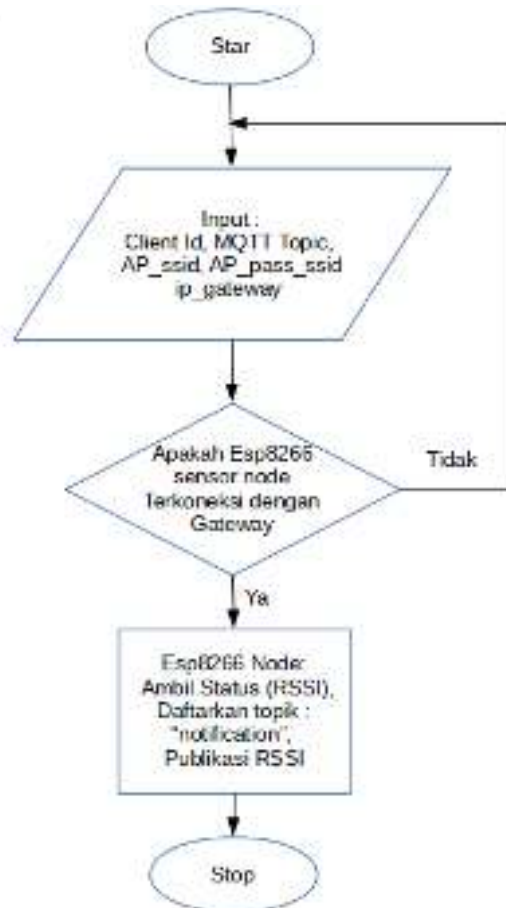
## METODOLOGI PENELITIAN

Penelitian ini terdiri atas dua bagian utama yaitu pengembangan pada sisi sensor node dan pengembangan pada sistem komunikasi mqtt untuk pengiriman data hasil pengukuran rssi sensor node. Gambar 1 memberikan informasi konfigurasi server mqtt pada gateway sedangkan gambar 2 memperlihatkan algoritma sensor node esp8266 mendapatkan status rssi dan publikasi ke gateway.

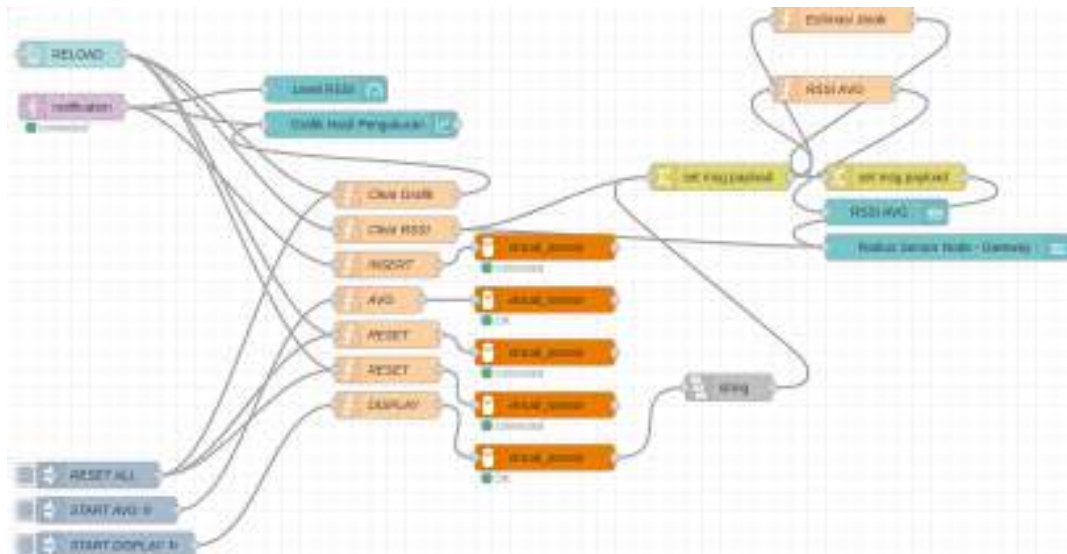
Terlihat pada gambar 2 inialisasi pengambilan data rssi pada sensor node esp8266 diawali dengan input data client, mqtt topic, ssid dan password akses point, serta ip gateway. Selanjutnya sensor node esp8266 akan melakukan pengecekan apakah sudah terkoneksi dengan akses point. Jika sensor node terkoneksi maka langkah selanjutnya adalah mendaftarkan topik publikasi sesuai



Gambar 1. Konfigurasi MQTT Server pada Gateway



Gambar 2. Algoritma Pengambilan Status RSSI dan Publikasi Ke Gateway



Gambar 3 RSSI Flow pada Gateway

Gambar 3 menjelaskan proses mendapatkan data publikasi rssi dari sensor node yang kemudian data tersebut tersimpan dalam databae mysql-server. Database virtual sensor pada gateway yang terlihat pada gambar 4 memiliki duah buah tabel yaitu data\_ukur dan ratarata. Tabel data\_ukur terlihat pada gambar 4 bertugas menyimpan data rssi hasil publikasi sensor node esp8266. Sementara itu tabel ratarata pada gambar 5 bertugas menyimpan proses perhitungan nilai mean rssi terukur.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	None		AUTO_INCREMENT
2	sinyal	int(5)			No	None		

Gambar 4. Struktur Tabel data\_ukur

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	None		AUTO_INCREMENT
2	rerata	int(11)			No	None		

Gambar 5. Struktur Tabel ratarata

Selain database terdapat tiga fungsi utama untuk melakukan pemrosesan data rssi pada sisi gateway yaitu fungsi "INSERT", fungsigi "AVG" dan fungsi "Estimasi Jarak". Fungsi "INSERT" pada gambar 6 bertugas untuk menginput data hasil publikasi sensor node si sisi gateway kedalam tabel data\_ukur. Sedangkan fungsi "AVG" pada gambar 7 bertujuan untuk mendapatkan

nilai mean rssi terukur tersebut. Fungsi "Estimasi Jarak" pada gambar 8 melakukan proses perhitungan estimasi jarak berdasarkan nilai mean rssi.

```
1 msg.topic = "INSERT INTO data_ukur(id, sinyal) VALUES ('+0+', '"+msg.payload+"');  
2 return msg;
```

Gambar 6. Fungsi "INSERT" pada RSSI Flow

```
1 msg.topic="INSERT INTO ratarata (rerata) SELECT AVG(sinyal) FROM data_ukur";  
2 return msg;
```

Gambar 8: Fungsi "AVG" pada RSSI Flow

```
1 msg.payload = Math.pow(10, (-35-msg.payload)/(10*2));  
2 return msg;
```

Gambar 7. Fungsi "Estimasi Jarak" pada RSSI Flow

Pengujian pada penelitian ini menggunakan dua buah perangkat IoT esp8266 dalam lingkungan propagasi LoS dimana salah satu berfungsi sebagai akses point dan lainnya berfungsi sebagai sensor node. Pada sisi sensor node kode programnya terlihat pada gambar 9 sedangkan akses point terlihat pada gambar 10.

```
import esp
esp.osdebug(None)
import time
from umqttsimple import MQTTClient
import machine
import micropython
import network
import ubinascii

client_id = ubinascii.hexlify(machine.unique_id())
topic_pub = b'notification'
ssid = 'esp'
password = '123456789'
mqtt_server = '192.168.4.2'
sta_if = network.WLAN(network.STA_IF) # Fungsi client
sta_if.active(True)
sta_if.connect('esp', '123456789')

akhir = 0
interval = 20
hitung = 0

client = MQTTClient(client_id, mqtt_server, user=b'root', password=b'esp')
client.connect()

while True:
    try:
        client.check_msg()
        if (time.time() - akhir) > interval:
            while sta_if.isconnected():
                pin = machine.Pin(2, machine.Pin.OUT)
                pin.value(0)
                time.sleep(0.5)
                pin.value(1)
                time.sleep(2)
                encrypted = str(sta_if.status('rssi'))
                print('signal_quality', sta_if.status('rssi'), 'dBm')
                msg = encrypted
                client.publish(topic_pub, msg)
                akhir = time.time()
                hitung += 1
            except OSError as e:
                restart_and_reconnect()
```

Gambar 9: Kode Program Sensor Node

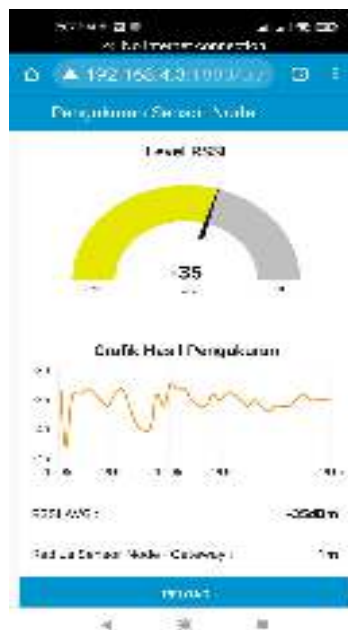
```
import network
import esp
esp.osdebug(None)
import gc
gc.collect()
ssid = 'esp'
password = '123456789'
ap = network.WLAN(network.AP_IF)
ap.active(True)
ap.config(essid=ssid, password=password)
print('Connection successful')
print(ap.ifconfig())
```

Gambar 10: Kode Program AP

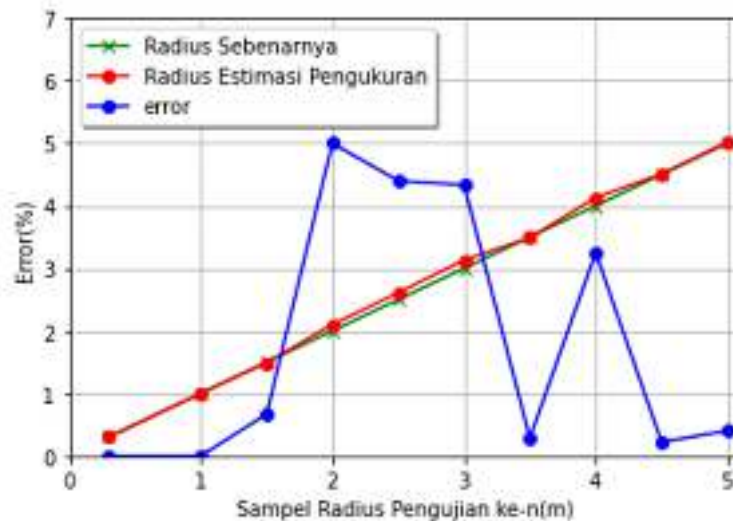
## HASIL DAN PEMBAHASAN

RSSI flow pada gambar 3 menghasilkan tampilan graphical user interface (GUI) yang terlihat pada Gambar 11. Gambar tersebut merupakan tampilan GUI yang dapat diakses oleh user dengan memasukkan alamat ip dari gateway disertai dengan memasukkan port 1880. seperti terlihat pada gambar 11 pengukuran radius untuk 1m memiliki nilai -35dBm.

Berdasarkan referensi nilai RSSI sebesar -35dBm pada jarak 1m tersebut maka hasil pengujian skala lab pada jarak antara 0.3m dan 5m dapat dilihat pada gambar 12. Terlihat pada gambar tersebut terdapat 100 sampel pengujian dengan error tertinggi pada jarak 2m sebesar 5% hingga 3m sebesar 4.4% dengan rata-rata error sebesar 1.85%



*Gambar 11 . GUI dari  
RSSI Flow*



Gambar 12. Hasil Pengujian dan Error

## KESIMPULAN

Penelitian ini berhasil mendapatkan estimasi radius sensor node terhadap gateway dalam kondisi LoS. Pengguna dapat melihat hasil pengujian melalui sebuah GUI ymelalui sebuah browser dengan memasukan alamat ip dari gateway disertai port komunikasi 1880. Hasil pengukuran pada kondisi LoS untuk jarak antara 0.3m hingga 5m memiliki rata-rata error sebesar 1.85%

## DAFTAR PUSTAKA

Chuku, N., & Nasipuri, A. (2021). RSSI-Based localization schemes for wireless sensor networks using outlier detection. *Journal of Sensor and Actuator Networks*, 10(1), 10.

ESP8266EX Datasheet Espressif.com, 2021." [Online]. Available: [https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf). [Accessed: 12 October 2021].

Jondhale, S. R., Sharma, M., Maheswar, R., Shubair, R., & Shelke, A. (2020). comparison of neural network training functions for rssi based indoor localization problem in WSN. In *Handbook of Wireless Sensor Networks: Issues and Challenges in Current Scenario's* (pp. 112-133). Springer, Cham.

Kurniawan, A. (2019). *Internet of Things Projects with ESP32: Build exciting and powerful IoT projects using the all-new Espressif ESP32*. Packt Publishing Ltd.

Node-RED. (2021). Retrieved 23 October 2021, from <https://nodered.org/>

Sicari, S., Rizzardi, A., & Coen-Porisini, A. (2019). Smart transport and logistics: A Node-RED



## **SEMINAR NASIONAL SAINS DAN TEKNIK FST UNDANA (SAINSTEK)**

Kupang, 02 November 2021

implementation. *Internet Technology Letters*,2(2), e88.

Vasilev, D. (2020, September). Python programming training with the robot Finch. In *2020 XXIX International Scientific Conference Electronics (ET)* (pp. 1-4). IEEE.