Optimizing Data Transfer Topologies in ESP-NOW Networks Using Minimum Spanning Tree (MST) Algorithms

Rivan Cahyadi - 13525007
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: rivancahyadi2018@gmail.com , 13525007@std.stei.itb.ac.id

Abstract— **The increasing deployment of Internet of Things (IoT) devices, such as ESP microcontrollers, within Wireless Sensor Networks (WSNs) requires efficient peer-to-peer communication protocols to reduce energy consumption and transmission latency. ESP-NOW provides a low-power Media Access Control (MAC) solution for these devices. However, modeling the network as a fully connected topology or Complete Graph may introduce inefficiencies when direct single-hop transmissions occur over weak, high-attenuation wireless links. This paper presents a graph-based method for optimizing ESP-NOW data transfer topologies using Minimum Spanning Tree (MST) algorithms. Prim's and Kruskal's algorithms are evaluated to compare the resulting routing paths, while Prim's algorithm is selected for embedded hardware execution due to its compatibility with adjacency matrices. By modeling the devices as graph vertices and the absolute Received Signal Strength Indicator (RSSI) as edge weights, this study constructs a routing tree with minimum total edge weight. Experimental results indicate that applying MST algorithms can reduce redundant transmission paths while maintaining connectivity among all nodes.**

Keywords—ESP-NOW, Minimum Spanning Tree, Prim Algorithm, Kruskal Algorithm, Graph Theory

I. INTRODUCTION

The Internet of Things (IoT) continues to grow rapidly, embedding smart devices ever more deeply into everyday life. In networks with limited resources, Wireless Sensor Networks (WSNs) depend on efficient peer-to-peer communication protocols to reduce energy usage and delays.

The ESP-NOW protocol, developed by Espressif Systems, provides an ultra-low-power, connectionless Media Access Control (MAC) layer solution tailored for decentralized IoT devices. It allows multiple ESP microcontrollers to communicate without the overhead of maintaining a Wi-Fi connection.

However, ESP-NOW natively operates on a flat topology where nodes default to direct, single-hop broadcasting. While a fully connected network (a Complete Graph) allows any device to communicate directly with another, relying on direct single-hop transmissions over varying physical distances poses significant physical and computational challenges. Forcing direct communication over weak wireless links characterized by high signal attenuation drastically increases packet loss and necessitates energy-consuming retransmissions.

To overcome this issue, data transfer topologies must be optimized. In Discrete Mathematics, graph theory provides robust solutions for network optimization. Specifically, the Minimum Spanning Tree (MST) concept can be utilized to find the most efficient routing paths, ensuring that data is forwarded only through the strongest available physical links without forming endless communication cycles.

This paper aims to examine a graph-based method for optimizing data transfer topologies in ESP-NOW networks using MST algorithms. By applying Prim's and Kruskal's algorithms to a practical setup of four ESP8266 modules, this study evaluates routing paths based on measured signal strength. The application of graph theory is expected to support the reduction of transmission redundancy in decentralized ESP-NOW networks.

II. LITERATURE REVIEW

A. Graph Theory

A graph $G = (V, E)$ is used to represent discrete objects and their relationships. V is a non-empty set of vertices (or nodes) $\{v_1, v_2, \dots, v_n\}$, and E is a set of edges $\{\mathrm{e}_1, \mathrm{e}_2, \dots, \mathrm{e}_\mathrm{n}\}$ that connect pairs of vertices. In the context of this study, the ESP8266 devices act as the vertices (V), and the wireless communication links between them act as the edges (E).

Graphs can be classified into several categories or types, depending on the perspective used for the classification. In general, graphs may be grouped based on the presence or absence of multiple edges or loops, as well as based on the orientation of their edges [1].

Based on the presence or absence of multiple edges or loops, graphs can be divided into two types: simple graphs and non-simple graphs. A simple graph contains neither loops nor multiple edges. In contrast, a non-simple graph contains either multiple edges (a multigraph) or loops (a pseudograph).

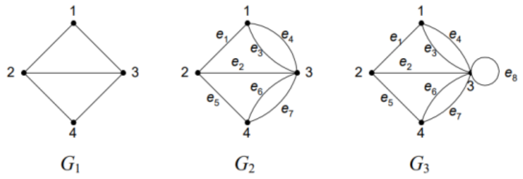


Fig 1. G_1 (Simple graph), G_2 (Multigraph), G_3 (Pseudograph) (Source: [1]).

In this study, the sensor network is modeled as a complete graph because each node lies within the radio-frequency range of every other node. This topology is denoted as a complete graph, designated as K_n. According to the Handshaking Lemma and fundamental graph theory, the number of edges in a complete graph is given by $|E| = \frac{|V|(|V|-1)}{2}$ [1]. Thus, for the four-node network analyzed in this paper, the topology forms a graph with exactly six bidirectional edges.

To evaluate the efficiency of communication across this topology, the network is further represented as a weighted graph, in which each edge is assigned a numerical value or weight. In this paper, the weight reflects the communication cost between two ESP8266 modules and is derived from the Received Signal Strength Indicator (RSSI).

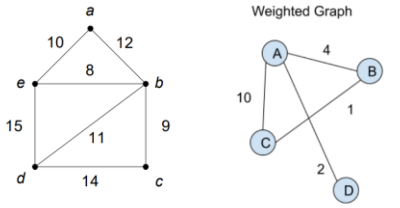


Fig 2. Illustration of Weighted Graph (Source: [1]).

B. Trees and Minimum Spanning Tree (MST) Algorithms

A tree is defined as a connected undirected graph with no simple circuits (cycles). For a graph with n vertices, a tree must contain exactly $n - 1$ edges. A spanning tree of a connected undirected graph G is a subgraph that is a tree and includes all the vertices of G [2].

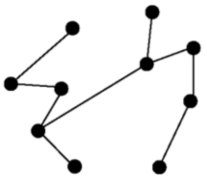


Fig 3. Example of Trees (Source []).

When a graph is weighted, it often has multiple spanning trees. A Minimum Spanning Tree (MST) is a spanning tree whose sum of edge weights is as small as possible. In this study, MST is used to determine the most efficient communication topology for connecting all ESP nodes with the minimum total transmission cost. This study evaluates two primary algorithms:

1) Prim's Algorithm

Developed by Robert Prim in 1957, this algorithm adopts a vertex-centric approach. It begins at an arbitrary root vertex and maintains a set of vertices already included in the MST [2]. In each iteration, it greedily selects the minimum-weight edge that connects a vertex within the MST to a vertex outside the MST, gradually expanding the tree until all vertices are incorporated. Prim's algorithm is commonly used for dense graphs, particularly when the graph is represented using an adjacency matrix.

2) Kruskal's Algorithm

Published by Joseph Kruskal in 1956, this algorithm adopts an edge-centric approach. It initially sorts all edges in the graph in non-decreasing order of their weights. It then iterates through the sorted list, adding edges to the MST provided they do not form a cycle. Kruskal's algorithm is generally suitable for sparse graphs where the number of edges is relatively low compared with the maximum possible connections.

C. Algorithmic Time Complexity

Asymptotic time complexity, denoted as $T(n)$, measures the number of computational steps performed by an algorithm as a function of the input size n. To evaluate algorithm performance for sufficiently large inputs, Big-O notation (O) is utilized to define the upper bound of the growth rate, formally expressed as $T(n) = O(f(n))$ [3]. In the context of resource-constrained embedded systems like the ESP8266, analyzing this asymptotic complexity is essential to prevent heap memory exhaustion or watchdog timer resets.

The execution time of MST algorithms depends on graph density and the data structures used in their implementation. When Prim's algorithm is implemented with an adjacency matrix, it searches for the minimum adjacent edge iteratively and has a quadratic time complexity of $O(|V^2|)$. In contrast, Kruskal's algorithm requires sorting all edges, resulting in a linearithmic time complexity of $O(|E| \, log \, |E|)$ or $O(|E| \, log \, |V|)$.

For dense complete graphs, such as K_n, where the number of edges approaches $|V|^2$, this complexity comparison is relevant for selecting an appropriate on-device execution strategy.

D. RSSI and Log-Distance Path Loss Modeling

To compute the Minimum Spanning Tree, a weight metric must be established for each edge. In wireless communications, the Received Signal Strength Indicator (RSSI) provides an empirical measurement of the power present in a received radio signal. RSSI is measured in decibels relative to a milliwatt (dBm) and can be used as an approximate indicator of link quality and relative spatial distance [4]. The propagation of radio waves in physical environments is commonly described using the Log-Distance Path Loss equation:

$$RSSI(d) = RSSI(d_0) - 10n \log_{10}\left(\frac{d}{d_0}\right) + X_\sigma \qquad (1)$$

In this model, the reference signal strength is measured at a known distance, the path loss exponent represents the attenuation characteristics of the environment; the distance term denotes the spatial separation between the transmitter and receiver; and the random variable accounts for multipath fading and shadowing effects. Because signal strength generally decreases with distance and environmental attenuation, long-distance direct transmissions may require higher transmission effort or experience lower reliability than shorter links under comparable conditions [5]. By using RSSI as the edge-weight metric, MST algorithms tend to prioritize links with lower measured attenuation, thereby producing a reduced set of communication paths for the evaluated topology.

E. *ESP-NOW Protocol*

Traditional Wi-Fi communication requires devices to associate with an Access Point, negotiate IP addresses via DHCP, and execute TCP handshakes. This process may introduce additional latency and energy consumption for constrained IoT devices. ESP-NOW reduces these requirements by operating at the Media Access Control (MAC) layer, enabling direct device-to-device communication without a conventional Wi-Fi association process [6].

ESP-NOW encapsulates its payloads within IEEE 802.11 Vendor-Specific Action Frames. Within the IEEE 802.11 standard, the Frame Control field identifies the packet as a management frame (Type 0x00) and an action frame (Subtype 0x0D or 0xd0). To distinguish ESP-NOW packets from ambient Wi-Fi traffic, the frame includes the Espressif Organizationally Unique Identifier (OUI), designated as 0x18, 0xfe, 0x34 [7].

While ESP-NOW supports Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA) to manage medium access, its raw transmission speed and lack of native routing algorithms necessitate the manual implementation of logical topologies to support multi-hop communications efficiently [8]. Furthermore, the maximum payload size of an ESP-NOW frame is strictly limited to 250 bytes, which forces routing tables and telemetry to be highly optimized and serialized before transmission.

III. METHODOLOGY AND SYSTEM ARCHITECTURE

A. *Topological Mapping of the Wireless Sensor Network*

The experimental hardware architecture comprises four ESP8266 microcontroller units operating as independent sensor nodes. In discrete mathematics, this network is formally mapped onto a weighted, undirected graph $G = (V, E)$. The set of vertices $V = \{A, B, C, D\}$, represents the microcontrollers. Due to the connectionless nature of the ESP-NOW Media Access Control (MAC) layer, where every node inherently possesses the capability to transmit data to all surrounding nodes within its radio range, the baseline topology forms a Complete Graph (K_4).

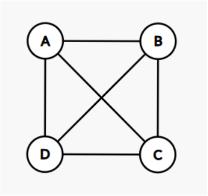


Fig 3. K_4 $Graph$ example.

In this configuration, every distinct pair of vertices is connected by a unique edge, resulting in a total of 6 bidirectional communication links. This dense mesh represents the maximum possible topological complexity for a four-node network.

B. *Experimental Setup and Spatial Distribution*

To evaluate the algorithmic response to physical signal attenuation, the four ESP8266 nodes were deployed at different physical distances and separated by selected structural obstacles. This spatial distribution provides a range of RSSI values for evaluating the proposed topology optimization approach under practical indoor conditions.

Node A was designated as the central root. Node C and Node D were placed in close physical proximity to Node A, both at a distance of 0.5 meters. Node B was positioned 4.5 meters away from Node A, representing a highly attenuated, obstructed wireless link. The distance between Node B and Node C was set to 4 meters with a clear Line-of-Sight (LoS). Node B and Node D were separated by 5 meters, while Node C and Node D were positioned 1 meter apart.

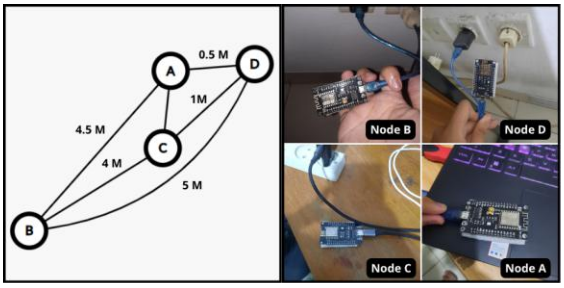


Fig 4. Physical spatial distribution of the ESP8266 nodes based on measured distances.

C. *RSSI-Based Edge Weighting*

To optimize the network, an edge weight function $W(e)$ must be established. The Received Signal Strength Indicator (RSSI), measured in decibel-milliwatts (dBm), serves as the primary metric for link quality and physical proximity.

In standard graph theory applications, edge weights typically denote non-negative distances or costs. RSSI values, conversely, are inherently negative integers, where a value closer to zero signifies minimal signal attenuation and closer physical proximity (e.g., -40 dBm is a significantly stronger link than -85 dBm). If raw RSSI values were directly applied to a Minimum Spanning Tree (MST) algorithm, the minimization objective would erroneously select the edges with the highest attenuation (since -85 < -40). To establish algorithmic coherence and ensure that the shortest physical distances correspond to the lowest mathematical costs, the weight function is defined using the absolute value of the RSSI:

$$W(e) = |RSSI| \tag{2}$$

Consequently, stronger measured signal quality corresponds to a lower numerical weight, allowing the MST algorithms to prioritize links with lower RSSI-based communication cost.

D. Distributed Adjacency Matrix Construction

To reduce the need for manual graph initialization, the nodes employ a distributed computational approach. Operating in the WIFI_AP_STA mode, each ESP8266 broadcasts its local proximity data through a dynamically encoded SSID while scanning for adjacent nodes. This concurrent operation enables the network to construct and populate an adjacency matrix representing the K4 graph. The matrix shown below is provided only as an illustrative example of the adjacency-matrix structure and does not represent the empirical values obtained from the practical experiment or hardware testing.

$$\begin{array}{ccccc} Dari/Ke & Node\ A & Node\ B & Node\ C & Node\ D \\ Node\ A & 0 & 42 & 65 & 81 \\ Node\ B & 42 & 0 & 55 & 60 \\ Node\ C & 65 & 55 & 0 & 40 \\ Node\ D & 81 & 60 & 40 & 0 \end{array} \tag{3}$$

E. Distributed Adjacency Matrix Construction

After the adjacency matrix is completed, the MST is computed locally on the designated central node’s embedded Xtensa 32-bit processor. The following C++ snippet shows the embedded Prim’s algorithm implementation, which processes the generated adjacency matrix with a time complexity of $O(|V^2|)$ to derive acyclic routing paths.

```
void executePrimMST(int graph[V][V]) {
   int parent[V];
   int key[V];
   bool mstSet[V];

   for (int i = 0; i < V; i++) {
        key[i] = 9999;
        mstSet[i] = false;
   }

   key[0] = 0;     // Initialize Node A as root
   parent[0] = -1;

   for (int count = 0; count < V - 1; count++) {
        int u = minKey(key, mstSet);
        mstSet[u] = true;

   for (int v = 0; v < V; v++) {
        if (graph[u][v] &&
            mstSet[v] == false &&
            graph[u][v] < key[v]) {

          parent[v] = u;
          key[v] = graph[u][v];
         }
       }
     }
   }
```

Although both Kruskal's and Prim's algorithms are theoretically evaluated in this study to validate the formation of the MST, their asymptotic time complexities dictate their feasibility for actual execution on the limited RAM of the ESP8266 microcontroller.

Kruskal's approach requires sorting all the edges of the graph. The standard sorting operation yields a linearithmic time complexity of $O(|E|\ log\ |E|)$. Furthermore, the sorting mechanism inherently demands dynamic memory operations and additional data structures (such as Union-Find) to detect cycles dynamically. This introduces significant processing overhead on resource-constrained microcontrollers.

Conversely, Prim's algorithm operates directly on the autonomously generated Adjacency Matrix. For a graph with $|V|$ vertices, it iteratively scans an array to find the minimum-weight adjacent vertex, resulting in a quadratic time complexity of $O(|V|\text{^}2)$.

For the dense K_4 complete graph generated by ESP-NOW ($|V| = 4$), the $O(|V|^2)$ complexity strictly bounds the execution 16 deterministic iterations. This constant, highly predictable loop circumvents the need for complex sorting and dynamic memory allocation.

Therefore, Prim's algorithm was exclusively selected to compute the routing tree locally with extreme efficiency on the central ESP8266 processor.

The complete C++ source code for the ESP8266 microcontrollers, including the distributed SSID scanning mechanism and the embedded Prim's algorithm execution, has been open-sourced and is accessible in the link: Click Here

IV. RESULTS AND DISCUSSION

A. Empirical Adjacency Matrix

After hardware deployment, the distributed scanning process generated a complete undirected adjacency matrix for the K_4 topology using the absolute RSSI value as the edge-weight metric.

The measurement results indicate that RSSI-based link quality is not solely determined by physical distance. The link between Node B and Node C produced the lowest edge weight, 42, although the nodes were separated by 4 meters, which can be attributed to the presence of a clear Line-of-Sight path. In contrast, the link between Node A and Node B, with a separation distance of 4.5 meters, produced the highest edge weight, 86, due to signal attenuation caused by physical obstructions. Furthermore, the link between Node A and Node C yielded a moderate edge weight of 62 despite the short 0.5-meter distance, suggesting the influence of near-field interference on the measured signal strength.

TABLE I. ADJACENCY MATRIX K_4 BASED ON THE |RSSI| METRIC

Node	**A**	**B**	**C**	**D**
A	0	86	62	59
B	86	0	42	61
C	62	42	0	55
D	59	61	55	0

Under the native ESP-NOW configuration (the unoptimized K_4 mesh state), the initial topological transmission cost is evaluated as $86 + 62 + 59 + 42 + 61 + 55 = 365$

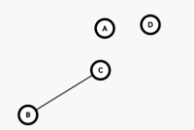
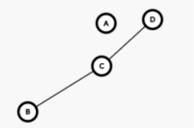
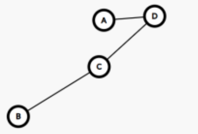
B. Tracing Kruskal's Algorithm

Kruskal's algorithm achieves optimization by sorting the graph's edges in non-decreasing order of their weights and iteratively selecting them if no cycles are formed.

Based on the measured edge weights, the edges are ordered in non-decreasing sequence as follows: $e_{BC}(42)$, $e_{CD}(55)$, $e_{AD}(59)$, $e_{BD}(61)$, $e_{AC}(62)$, and $e_{AB}(86)$.

Using this ordered sequence, the construction of the MST can be traced from the first accepted edge to the final spanning tree. The detailed selection results, including the accepted and rejected edges at each iteration, are presented in Table 2.

TABLE II. KRUSKAL'S ALGORITHM ITERATION PROCESS

Iteration	Evaluated Edge	Weight	Decision	Spanning Tree State
1	B-C	42	Accepted	
2	C-D	55	Accepted	
3	A-D	59	Accepted	
4	B-D	61	Rejected	Cycle formed
5	A-C	62	Rejected	Cycle formed
6	A-B	86	Rejected	Cycle formed

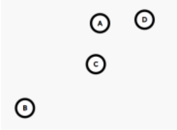
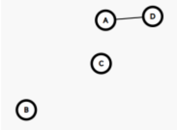
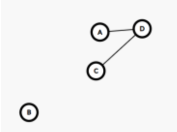
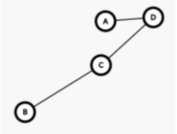
As shown in Table 2, the first three selected edges, namely $e_{BC}(42)$, $e_{CD}(55)$, $e_{AD}(59)$, satisfy the acyclic property and connect all four vertices. Since a spanning tree with four vertices requires exactly three edges, the MST construction is completed after the third iteration. The remaining edges are rejected because their inclusion would introduce cycles into the established tree structure. Therefore, the resulting MST consists of {(B,C), (C,D), (A,D)} with a total edge weight of 156.

C. Tracing Prim's Algorithm

Prim's algorithm constructs the MST dynamically by applying the cut property. Starting from a designated root vertex, the tree is expanded by selecting the minimum-weight incident edge that connects the current tree set to an unvisited vertex. In this tracing process, vertices enclosed in brackets indicate the nodes currently included in the tree set.

Node A is selected as the initial root vertex, so the initial tree set is $V_T = \{A\}$. The iteration process then evaluates the minimum connecting edge from the current tree set to the remaining unvisited vertices, as summarized in Table 3.

TABLE III. PRIM'S ALGORITHM ITERATION PROCESS

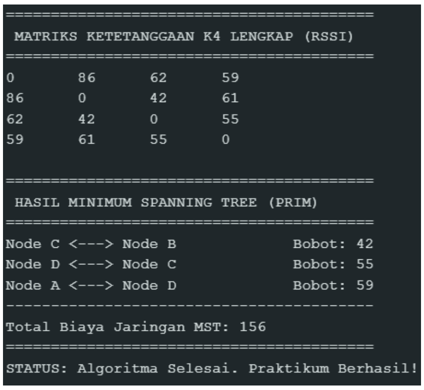
Iteration	Edge	Weight	Spanning Forest
Initialization	-	-	
1	A-D	59	
2	D-C	55	
3	C-B	42	

As shown in Table 3, Prim's algorithm expands the spanning forest from the initial root node by successively selecting A-D, D-C, and C-B. After the third step, all vertices are contained in a single spanning tree. The resulting total transmission cost is $59 + 55 + 42 = 156$, which is identical to the MST obtained using Kruskal's algorithm.

D. Optimization Analysis and Hardware Validation

The mathematical traces of Kruskal's and Prim's algorithms produced the same minimum spanning tree for the evaluated topology.

To validate the implementation, Prim's algorithm was executed on the embedded Xtensa processor of the central ESP8266 node. As shown in Figure 5, the hardware generated the adjacency matrix and computed the same MST branches and total cost of 156 as the manual mathematical trace.



```
========================================
 MATRIKS KETETANGGAAN K4 LENGKAP (RSSI)
========================================
0       86      62      59
86      0       42      61
62      42      0       55
59      61      55      0


========================================
 HASIL MINIMUM SPANNING TREE (PRIM)
========================================
Node C <---> Node B            Bobot: 42
Node D <---> Node C            Bobot: 55
Node A <---> Node D            Bobot: 59
----------------------------------------
Total Biaya Jaringan MST: 156
========================================
STATUS: Algoritma Selesai. Praktikum Berhasil!
```

Fig 5. Serial monitor output showing the on-device execution of Prim's algorithm.

By restricting the ESP-NOW communication protocol to forward packets exclusively along these computed MST edges through a unicast mechanism, the topological transmission cost was reduced from the initial 365 to 156.

This result corresponds to a 57.26% reduction in the evaluated topological cost. The empirical data also illustrates a practical networking condition in which the direct 4.5-meter link between Node A and Node B produced the highest weight, while the MST selected a multi-hop route through lower-weight links. Furthermore, reducing the communication edges from 6 to 3 removes cycles from the selected topology, which can help limit redundant transmissions in the evaluated ESP-NOW configuration.

V. CONCLUSION

This study demonstrates the application of discrete mathematics for analyzing and optimizing an embedded wireless communication topology. By mapping an ESP-NOW micro-network to a Complete Graph (K4) and transforming negative RSSI values into absolute edge weights, the flat wireless topology can be represented as a weighted graph for MST-based evaluation.

The hardware experiment showed variations in signal propagation caused by distance, obstacles, and local environmental conditions. The mathematical traces of Kruskal's and Prim's algorithms produced the same MST for the evaluated graph, with a total edge weight of 156 and a 57.26% reduction in the evaluated topological cost compared with the complete graph. For the implemented embedded system, Prim's algorithm was selected because it operates directly on the adjacency matrix generated by the nodes. Overall, the results indicate that MST-based topology selection can reduce redundant communication paths and provide an acyclic unicast structure for the tested ESP-NOW network

VIDEO LINK AT YOUTUBE

Link to the video of the trial results on YouTube: Click Here

ACKNOWLEDGMENT

The author would like to express profound gratitude to God Almighty for His grace. Sincere appreciation is extended to Prof. Dr. Ir. Rinaldi, M.T., for his continuous guidance and comprehensive teaching of the IF1220 Matematika Diskrit course, which provided the essential foundational knowledge implemented and analyzed in this paper.

REFERENCES

[1] R. Munir, "Graf (Bagian 1)," *Bahan Kuliah IF1220 Matematika Diskrit*, Program Studi Teknik Informatika, STEI-ITB, 2026. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2025-2026/20-Graf-Bagian1-2026.pdf. (Diakses: 19 Juni 2026).

[2] R. Munir, "Pohon (Bagian 1)," *Bahan Kuliah IF1220 Matematika Diskrit*, Program Studi Teknik Informatika, STEI-ITB, 2026. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2025-2026/23-Pohon-Bag1-2026.pdf. (Diakses: 19 Juni 2026).

[3] R. Munir, "Kompleksitas Algoritma (Bagian 1)," *Bahan Kuliah IF1220 Matematika Diskrit*, Program Studi Teknik Informatika, STEI-ITB, 2026. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2025-2026/25-Kompleksitas-Algoritma-Bagian1-2026.pdf. (Diakses: 19 Juni 2026).

[4] "Estimate distance measurement using NodeMCU ESP8266 based on RSSI technique," ResearchGate. [Online]. Available: https://www.researchgate.net/publication/322877438_Estimate_distance_measurement_using_NodeMCU_ESP8266_based_on_RSSI_technique. (Diakses: 19 Juni 2026).

[5] "Extending Lifetime of Sensor Networks Based on Sleep-scheduled Routing Algorithm," Research Commons. [Online]. Available: https://dc-china-simulation.researchcommons.org/cgi/viewcontent.cgi?article=2181&context=journal. (Diakses: 19 Juni 2026).

[6] Espressif Systems, "ESP-NOW: Espressif's Wireless Communication Protocol," Espressif News. [Online]. Available: https://www.espressif.com/en/news/ESP-NOW. (Diakses: 19 Juni 2026).

[7] "ESP-NOW protocol measure receiver power strength of signal strength," ESP32 Forum. [Online]. Available: https://esp32.com/viewtopic.php?t=13889. (Diakses: 19 Juni 2026).

[8] "A Real-World Evaluation of ESP32's Suitability for Use in Wireless Ad Hoc Networks," Digitalbibliothek Thüringen. [Online]. Available: https://www.db-thueringen.de/servlets/MCRFileNodeServlet/dbt_derivate_00068872/ilm1-202520021_009-012.pdf. (Diakses: 19 Juni 2026).

STATEMENT

I hereby declare that the paper I wrote is my own writing, not an adaptation or translation of someone else's paper, and is not plagiarized.

Bandung, 1 Juni 2025

Rivan Cahyadi - 13525007