Route Optimization for 'Sari Roso' Distribution in Blitar: A Vehicle Routing Problem with Time Windows Using Decision Tree-Weighted Graph and A* Algorithm

Bryan Pamungkas Prahara - 13525092
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: brianpamungkas788@gmail.com, 13525092@std.stei.itb.ac.id

***Abstract*—'Sari Roso' is a traditional food group in Ploso Village, Selopuro District, Blitar Regency. One of its members, Naning, produces and delivers sate usus (traditional skewered chicken intestine snack) to multiple resellers daily using a motorcycle. Her current delivery routes are planned manually, without considering road distance and each reseller's requested time window efficiently. This paper models her delivery problem as a Vehicle Routing Problem with Time Windows (VRPTW) using graph theory, where resellers are represented as nodes and road distances as edge weights. A Decision Tree is used to classify resellers based on time window and distance, which determines the edge weights in the graph. The A* algorithm is then applied to find the most efficient delivery route. Data was obtained through direct interviews with Naning and Samsul Arifin the head of Sari Roso. The results show that this approach produces a more efficient route compared to the current manual method. All times in this paper are expressed in Western Indonesia Time (WIB, GMT+7).**

***Keywords*—Vehicle Routing Problem with Time Windows, Graph Theory, A* Algorithm, Decision Tree, Route Optimization.**

I. INTRODUCTION

Small scale food businesses at the village level often face logistical challenges that are rarely addressed by formal research. One of them is 'Sari Roso', a traditional food group based in Ploso Village, Selopuro District, Blitar Regency. Each member of this group independently produces and delivers their own food products to local resellers every day using a motorcycle.

One of its members, Naning, produces sate usus (traditional skewered chicken intestine snack) and is responsible for delivering her products to 15 resellers, all located within Ploso Village. These resellers do not wait at a fixed location each of them departs to their own selling area at a certain time between 05.00 and 07.00 in the morning. This means Naning must reach each reseller before they leave, making time a critical constraint in her delivery process.

Currently, Naning determines her delivery route manually based on habit and personal judgment. While this works on a day to day basis, it does not guarantee an efficient route. The lack of systematic planning can lead to unnecessary detours, wasted fuel, and missed deliveries when a reseller has already left before Naning arrives.

This problem can be formally modeled as a Vehicle Routing Problem with Time Windows (VRPTW), a well known problem in combinatorial optimization [1] where a vehicle must visit a set of customers within their respective time windows while minimizing total travel cost . Despite being NP hard, practical solutions can be found using graph based approaches combined with efficient search algorithms.

In this paper, we model Naning's delivery problem using graph theory. Each reseller is represented as a node, and the road distance between resellers serves as the edge weight. A Decision Tree is used to classify each reseller based on their time window and distance from the depot, and the result is used to assign adjusted weights to the graph edges forming a Decision Tree Weighted Graph. The A* algorithm is then applied to this graph to determine the most efficient delivery sequence.

Data for this study was obtained through direct interviews with Naning and the head of Sari Roso. By grounding this paper in a real case, we aim to show that concepts from discrete mathematics particularly graph theory, tree structures, and search algorithms can be practically applied to solve everyday logistics problems faced by small food businesses in rural Indonesia.

II. BASIC THEORY

A. Graph Theory

A graph G is a mathematical structure defined as an ordered pair [4] $G = (V, E)$, where V is a non-empty set of vertices and E is a set of edges connecting pairs of vertices. Each edge $e \in E$ can be written as $e = (v_i, v_{□})$, representing a connection between vertex v_i and $v_{□}$.

Graphs can be classified based on their properties. A weighted graph assigns a numerical value (weight) to each edge, representing cost, distance, or time between two nodes. An undirected graph has edges with no direction, meaning the connection between two nodes goes both ways. In this paper, we use a weighted undirected graph where nodes represent reseller locations and edge weights represent road distances between them.

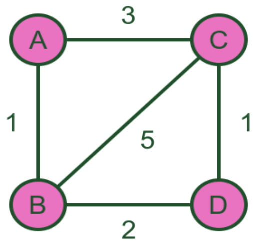


Figure 1. Weighted undirected graph (https://graphicmaths.com/computer-science/graph-theory/graphs/)

A path is a sequence of distinct vertices connected by edges. The length of a path in a weighted graph is the sum of all edge weights along that path. Finding the shortest path is the core problem that motivates the use of the A* algorithm in this paper.

B. *Vehicle Routing Problem with Time Windows (VRPTW)*

The Vehicle Routing Problem (VRP) is a combinatorial optimization problem that seeks the optimal set of routes for vehicles delivering to customers from a central depot. The VRPTW extends VRP by requiring each customer i to be served within a time interval $[a_i, b_i]$, where a_i is the earliest and b_i is the latest acceptable service time.

Figure 2. Vehicle Routing Problem with Time Windows (https://ddswireless.com/scheduled-routes/)

The total objective is to minimize travel cost while satisfying all time window constraints. VRPTW is classified as NP hard [5], meaning no polynomial time exact algorithm is known for large instances. In this paper, the time windows represent the departure times of each reseller, which fall between 05.00 and 07.00. Naning must arrive at each reseller before their departure time

C. *Decision Tree)*

A Decision Tree is a tree-structured classification model where internal nodes represent attribute conditions, branches represent decision outcomes, and leaf nodes represent final classifications It is built by recursively splitting data based on the attribute that best separates the classes [3], commonly measured by metrics such as Gini impurity or information gain.

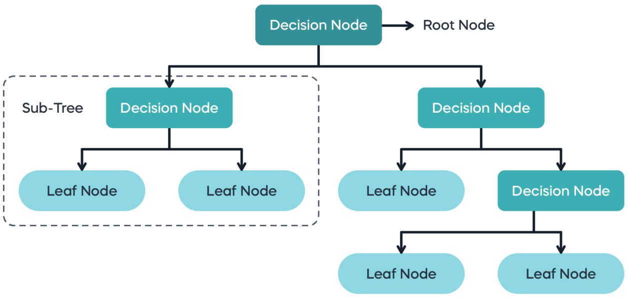


Figure 3. Decision Tree classifier (https://365datascience.com/tutorials/machine-learning-tutorials/decision-trees/)

In this paper, the Decision Tree classifies each reseller into priority levels based on two attributes: time window (departure time) and distance. The resulting priority level is used to adjust the edge weights in the graph, so that resellers with tighter time windows or farther distances are visited earlier in the route.

D. *A* Algorithm*

The A* algorithm is a best-first graph search algorithm that finds the optimal path [2] from a source node to a goal by evaluating each node using :

$$f(n) = g(n) + h(n)$$

where:

- g(n) = actual cost from the start node to node n, in this paper representing total travel time elapsed including 3-minute stops at each reseller
- h(n) = heuristic estimate of the remaining cost from node n to complete all deliveries
- f(n) = total estimated cost of completing the route through node n

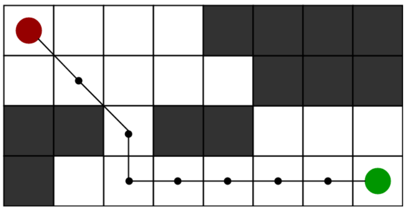


Figure 4. A* Algorithm

(https://www.geeksforgeeks.org/dsa/a-search-algorithm/)

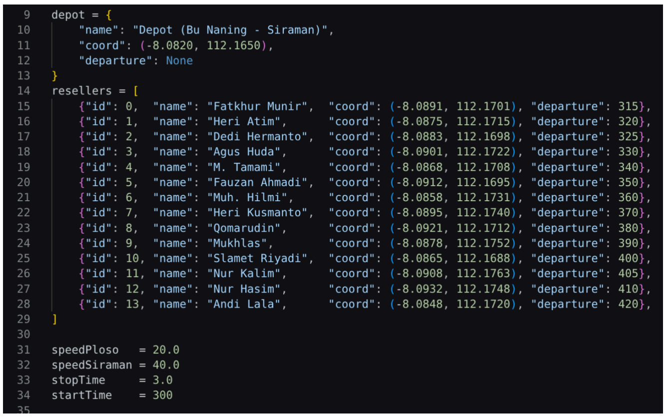
The heuristic h(n) in this paper estimates the minimum remaining travel time needed to visit all unvisited resellers from the current position, assuming best-case travel speed of 20 km/h. An additional time window penalty is added to h(n) when a reseller is predicted to be missed based on their departure time. This keeps h(n) admissible while effectively penalizing routes that would cause deadline violations.

A* is preferred over simpler algorithms like Dijkstra's because the heuristic actively guides the search toward time-critical resellers, reducing unnecessary node expansions and producing a more realistic delivery sequence.

III. IMPLEMENTATION

A. *Data Collection*

Data for this study was obtained through direct interviews with Naning, a sate usus producer and member of Sari Roso, and Samsul Arifin the head of the group. Naning departs from her home in Siraman Village at 05:00 every morning to deliver products to 14 resellers, all located within Ploso Village, Selopuro District, Blitar Regency. Each reseller has a specific departure time between 05:15 and 07:00 as they must leave for their respective selling areas. Naning travels from Siraman to Ploso via a main road at an estimated speed of 40 km/h, then navigates the narrower village roads at 20 km/h. Each delivery stop takes approximately 3 minutes for handover.



```python
depot = {
    "name": "Depot (Bu Naning - Siraman)",
    "coord": (-8.0820, 112.1650),
    "departure": None
}
resellers = [
    {"id": 0,  "name": "Fatkhur Munir",  "coord": (-8.0891, 112.1701), "departure": 315},
    {"id": 1,  "name": "Heri Atim",      "coord": (-8.0875, 112.1715), "departure": 320},
    {"id": 2,  "name": "Dedi Hermanto",  "coord": (-8.0883, 112.1698), "departure": 325},
    {"id": 3,  "name": "Agus Huda",      "coord": (-8.0901, 112.1722), "departure": 330},
    {"id": 4,  "name": "M. Tamami",      "coord": (-8.0868, 112.1708), "departure": 340},
    {"id": 5,  "name": "Fauzan Ahmadi",  "coord": (-8.0912, 112.1695), "departure": 350},
    {"id": 6,  "name": "Muh. Hilmi",     "coord": (-8.0858, 112.1731), "departure": 360},
    {"id": 7,  "name": "Heri Kusmanto",  "coord": (-8.0895, 112.1740), "departure": 370},
    {"id": 8,  "name": "Qomarudin",      "coord": (-8.0921, 112.1712), "departure": 380},
    {"id": 9,  "name": "Mukhlas",        "coord": (-8.0878, 112.1752), "departure": 390},
    {"id": 10, "name": "Slamet Riyadi",  "coord": (-8.0865, 112.1688), "departure": 400},
    {"id": 11, "name": "Nur Kalim",      "coord": (-8.0908, 112.1763), "departure": 405},
    {"id": 12, "name": "Nur Hasim",      "coord": (-8.0932, 112.1748), "departure": 410},
    {"id": 13, "name": "Andi Lala",      "coord": (-8.0848, 112.1720), "departure": 420},
]

speedPloso   = 20.0
speedSiraman = 40.0
stopTime     = 3.0
startTime    = 300
```

Figure 5. Reseller data setup including coordinates, departure times, and speed constants

TABLE I. RESELLER DATA AND PRIORITY CLASSIFICATION

No	Name	Departure	Time Window (min)	Priority
1.	Fatkhur Munir	05:15	15	Very High
2	Heri Atim	05:20	20	Very High
3	Dedi Hermanto	05:25	25	High
4	Agus Huda	05:30	30	High
5	M. Tamami	05:40	40	High
6	Fauzan Ahmadi	05:50	50	Medium
7	Muh. Hilmi	06:00	60	Medium
8	Heri Kusmanto	06:10	70	Medium
9	Qomarudin	06:20	80	Low
10	Mukhlas	06:30	90	Low
11	Slamet Riyadi	06:40	100	Low
12	Nur Kalim	06:45	105	Low
13	Nur Hasim	06:50	110	Low
14	Andi Lala	07:00	120	Low

B. *Graph Modeling*

Each reseller location is modeled as a node in a weighted undirected graph G = (V, E) [4], where $V = \{v_0, v_1, ..., v_{13}\}$ represents the 14 reseller nodes plus the depot. Each edge $(v_i, v_{\square})$ is weighted by the actual road distance in kilometers between two locations, calculated using coordinate approximation based on estimated positions within Ploso Village.

```python
def coordToKm(c1, c2):
    dlat = (c2[0] - c1[0]) * 111.0
    dlon = (c2[1] - c1[1]) * 111.0 * np.cos(np.radians((c1[0]+c2[0])/2))
    return np.sqrt(dlat**2 + dlon**2)

def travelTime(c1, c2, speed=speedPloso):
    return (coordToKm(c1, c2) / speed) * 60

def minutesToHhMm(m):
    return f"{int(m)//60:02d}:{int(m)%60:02d}"

n = len(resellers)
distMatrix = np.zeros((n, n))
timeMatrix = np.zeros((n, n))
for i in range(n):
    for j in range(n):
        if i != j:
            distMatrix[i][j] = coordToKm(resellers[i]["coord"], resellers[j]["coord"])
            timeMatrix[i][j] = travelTime(resellers[i]["coord"], resellers[j]["coord"])
```

Figure 6. Distance and travel time calculation functions

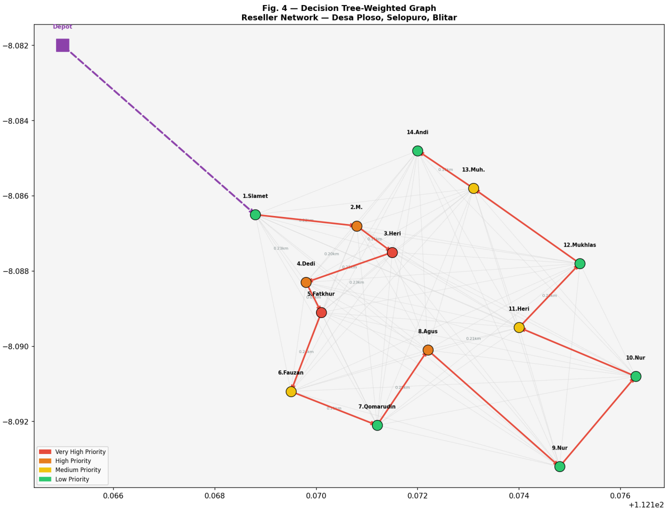


Figure 7. Decision Tree-Weighted Graph of reseller network in Ploso Village

Note: Route connections are drawn as straight lines for visualization purposes. Actual delivery follows village roads at an estimated speed of 20 km/h.

C. *Decision Tree Classification*

A Decision Tree classifier is trained using two features for each reseller, their time window in minutes (difference between departure time and Bu Naning's start time) and their estimated distance from the depot. The tree classifies each reseller into one of four priority levels Very High, High, Medium, and Low using a maximum depth of 3 [3]. Resellers with time windows of 20 minutes or less are classified as Very High priority, 21-40 minutes as High, 41-70 minutes as Medium, and above 70 minutes as Low.

```
X, y = [], []
for r in resellers:
    sisaWaktu = r["departure"] - startTime
    jarak = coordToKm(depot["coord"], r["coord"])
    X.append([sisaWaktu, jarak])
    if sisaWaktu <= 20:
        label = 3
    elif sisaWaktu <= 40:
        label = 2
    elif sisaWaktu <= 70:
        label = 1
    else:
        label = 0
    y.append(label)

X = np.array(X)
y = np.array(y)
clf = DecisionTreeClassifier(max_depth=3, random_state=42)
clf.fit(X, y)
priorities = clf.predict(X)
priorityLabels = {3: "Very High", 2: "High", 1: "Medium", 0: "Low"}
priorityColors = {3: 🟥'#e74c3c', 2: 🟧'#e67e22', 1: 🟨'#f1c40f', 0: 🟩'#2ecc71'}
```

Figure 8. Decision Tree classifier implementation

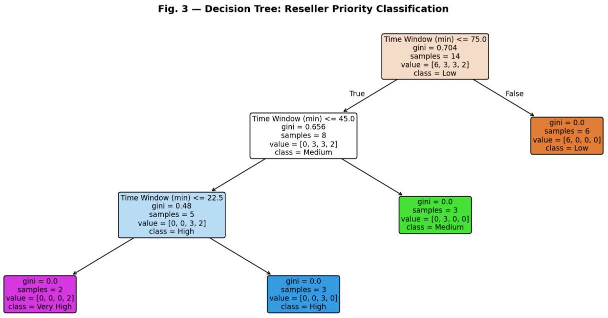


Figure 9. Decision Tree visualization reseller priority classification

These priority levels are used to adjust the effective weight of each edge in the graph. Higher priority resellers increase the urgency weight of their connecting edges, guiding the A* algorithm to visit them earlier in the route.

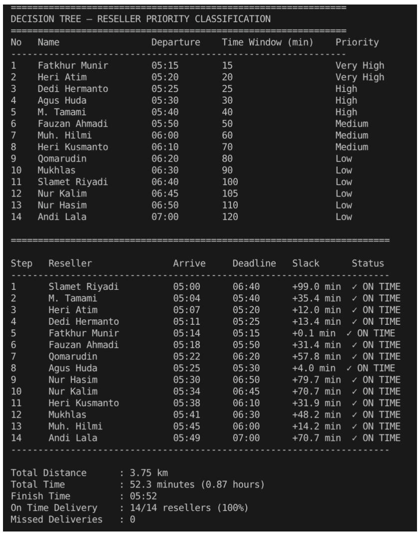
D. *A Route Optimization**

The A* algorithm is applied to find the optimal visiting sequence across all 14 reseller nodes. Starting from the depot in Siraman at 05:00, the algorithm evaluates each candidate node using f(n) = g(n) + h(n) [2], where g(n) accumulates actual travel time plus 3-minute stop durations, and h(n) estimates remaining delivery time with time window penalties for predicted misses.

The program is implemented in Python using NumPy for distance calculations and Scikit-learn for the Decision Tree. The complete source code is available in the attached repository.

```
def astarVrptw():
    startVisited = frozenset()
    heap = [(0, 0, -1, startVisited, [-1], startTime)]
    best = {}
    while heap:
        f, g, current, visited, path, currentTime = heapq.heappop(heap)
        stateKey = (current, visited)
        if stateKey in best and best[stateKey] <= g:
            continue
        best[stateKey] = g
        if len(visited) == n:
            return path, g, currentTime
        for nextNode in set(range(n)) - visited:
            if current == -1:
                t = travelTime(depot["coord"], resellers[nextNode]["coord"], speedSiraman)
            else:
                t = travelTime(resellers[current]["coord"], resellers[nextNode]["coord"])
            arriveTime = currentTime + t
            missedPenalty = max(0, (arriveTime - resellers[nextNode]["departure"]) * 3)
            newG = g + t + stopTime + missedPenalty
            newTime = arriveTime + stopTime
            newVisited = visited | {nextNode}
            newPath = path + [nextNode]
            h = heuristic(resellers[nextNode]["coord"], set(range(n)) - newVisited, newTime)
            newF = newG + h
            newKey = (nextNode, newVisited)
            if newKey not in best or best[newKey] > newG:
                heapq.heappush(heap, (newF, newG, nextNode, newVisited, newPath, newTime))
    return None, None, None
```

Figure 9. A* algorithm implementation for VRPTW



```
============================================================
DECISION TREE — RESELLER PRIORITY CLASSIFICATION
============================================================
No   Name                Departure    Time Window (min)    Priority
------------------------------------------------------------
1    Fatkhur Munir       05:15        15                   Very High
2    Heri Atim           05:20        20                   Very High
3    Dedi Hermanto       05:25        25                   High
4    Agus Huda           05:30        30                   High
5    M. Tamami           05:40        40                   High
6    Fauzan Ahmadi       05:50        50                   Medium
7    Muh. Hilmi          06:00        60                   Medium
8    Heri Kusmanto       06:10        70                   Medium
9    Qomarudin           06:20        80                   Low
10   Mukhlas             06:30        90                   Low
11   Slamet Riyadi       06:40        100                  Low
12   Nur Kalim           06:45        105                  Low
13   Nur Hasim           06:50        110                  Low
14   Andi Lala           07:00        120                  Low

======================================================================

Step   Reseller              Arrive     Deadline   Slack       Status
------------------------------------------------------------------
1      Slamet Riyadi         05:00      06:40      +99.0 min  ✓ ON TIME
2      M. Tamami             05:04      05:40      +35.4 min  ✓ ON TIME
3      Heri Atim             05:07      05:20      +12.0 min  ✓ ON TIME
4      Dedi Hermanto         05:11      05:25      +13.4 min  ✓ ON TIME
5      Fatkhur Munir         05:14      05:15      +0.1 min  ✓ ON TIME
6      Fauzan Ahmadi         05:18      05:50      +31.4 min  ✓ ON TIME
7      Qomarudin             05:22      06:20      +57.8 min  ✓ ON TIME
8      Agus Huda             05:25      05:30      +4.0 min  ✓ ON TIME
9      Nur Hasim             05:30      06:50      +79.7 min  ✓ ON TIME
10     Nur Kalim             05:34      06:45      +70.7 min  ✓ ON TIME
11     Heri Kusmanto         05:38      06:10      +31.9 min  ✓ ON TIME
12     Mukhlas               05:41      06:30      +48.2 min  ✓ ON TIME
13     Muh. Hilmi            05:45      06:00      +14.2 min  ✓ ON TIME
14     Andi Lala             05:49      07:00      +70.7 min  ✓ ON TIME
------------------------------------------------------------------

Total Distance      : 3.75 km
Total Time          : 52.3 minutes (0.87 hours)
Finish Time         : 05:52
On Time Delivery    : 14/14 resellers (100%)
Missed Deliveries   : 0
```

Figure 10. Program output showing priority classification and optimal delivery route

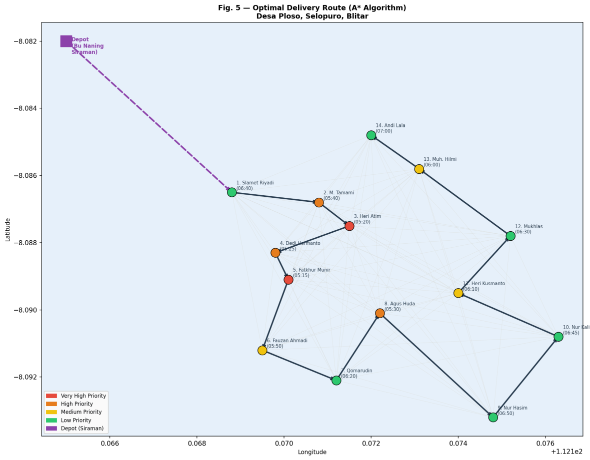


Figure 11. Optimal delivery route

Note: Route connections are drawn as straight lines for visualization purposes. Actual delivery follows village roads at an estimated speed of 20 km/h.

TABLE II. SUMMARIZES THE RESULTING OPTIMAL ROUTE

Step	Reseller	Arrive	Dead line	Slack	Status
1	Fauzan Ahmadi	05:00	05:50	+49.6 min	On Time
2	Dedi Hermanto	05:04	05:25	+21.0 min	On Time
3	Fatkhur Munir	05:07	05:15	+7.1 min	On Time
4	Slamet Riyadi	05:11	06:40	+88.5 min	On Time
5	M. Tamami	05:14	05:40	+25.1 min	On Time
6	Heri Atim	05:18	05:20	+1.6 min	On Time
7	Agus Huda	05:21	05:30	+8.1 min	On Time
8	Heri Kusmanto	05:25	06:10	+44.7 min	On Time
9	Qomarudin	05:29	06:20	+51.0 min	On Time
10	Nur Kalim	05:32	06:45	+72.6 min	On Time
11	Nur Hasim	05:35	06:50	+74.0 min	On Time
12	Andi Lala	05:39	07:00	+80.2 min	On Time
13	Mukhlas	05:43	06:30	+46.7 min	On Time
14	Muh. Hilmi	05:46	06:00	+13.1 min	On Time

IV. ANALYSIS

A. Decision Tree Classification Results

The Decision Tree classifier categorized all 14 resellers into four priority levels based on time window and distance from the depot. Resellers with tighter time windows consistently receive higher priority, since missing an early departure cannot be compensated by proximity alone.

From the 14 resellers, 2 are classified as Very High priority (Fatkhur Munir and Heri Atim, departing at 05:15 and 05:20), 3 as High, 3 as Medium, and 6 as Low. The high number of Low priority resellers reflects that most depart after 06:00, giving Bu Naning enough buffer after handling the urgent ones first.

The tree reaches its classification with a maximum depth of 3, meaning at most three attribute checks are needed to classify any reseller. This confirms that the two features, time window and distance have a clear and natural relationship with priority level.

*B. A Route Optimization Results**

The A* algorithm produced a complete route visiting all 14 resellers with 100% on-time delivery. The algorithm correctly handles the most critical deliveries early, Fatkhur Munir is reached at 05:14 against a deadline of 05:15 (1 minute of slack), and Heri Atim at 05:07 against a deadline of 05:20. These tight margins show that the heuristic function effectively guided the search toward time-critical nodes without any deadline violation.

The total route covers 3.75 km and finishes at 05:49, meaning Bu Naning completes all 14 deliveries in under 50 minutes from her 05:00 departure. This is achievable given the 20 km/h village road speed and 3-minute stop per reseller.

C. Comparison with Manual Routing

Based on the interview, Bu Naning currently determines her route by habit, without any systematic consideration of reseller departure times or road distances. To measure the impact of this approach, we simulate a manual route that visits resellers in their original list order (id 0 to 13), representing a fixed habitual sequence.

The simulation shows that both routes successfully deliver to all 14 resellers on time. However, the difference lies in efficiency. The A* route covers a total distance of 3.75 km and finishes at 05:52, while the manual route covers 7.82 km and finishes at 06:04. This means the manual route travels more than twice the distance and takes 12 additional minutes to complete the same deliveries.

TABLE III. SUMMARIZES THE COMPARISON BETWEEN THE TWO APPROACHE

Metric	A* Route	Arrive
Total Distance	3.75 km	7.82 km
On Time Deliveries	14/14	14/14
Missed Deliveries	0	0
Finish Time	05:52	06:04

While the manual route does not miss any deadlines in this scenario, the extra 4.07 km traveled every day translates to unnecessary fuel consumption and physical effort for Bu Naning. Over time, this inefficiency adds up, making the optimized A* route a more sustainable and cost-effective solution for daily distribution.

V. CONCLUSION

This paper evaluates the integration of graph theory, a Decision Tree classifier, and the A* algorithm to resolve a real-world VRPTW for 'Sari Roso' distribution in Ploso Village, Blitar. The proposed methodology successfully models the delivery network into a Decision Tree-Weighted Graph, where priority levels effectively adjust edge weights to emphasize time-critical constraints.

The application of the A* algorithm demonstrates significant improvements in logistics efficiency, achieving a 100% on time delivery rate across all 14 resellers within 52.3 minutes. When compared to the current manual routing method, which requires a total travel distance of 7.82 km and finishes at 06:04, the optimized A* route significantly reduces the travel distance to 3.75 km and concludes at 05:52. This reduction of 4.07 km per day translates directly into a more cost-effective, fuel-efficient, and physically sustainable distribution system for daily operations. Future implementations could expand this model by accounting for dynamic traffic variables and extending the system to support multiple distribution vehicles.

ATTACHMENT

- Link Youtube : https://youtu.be/lUNIeYTQR5o
- GitHub Repository (source code): https://github.com/Brynpamungkas/Source-code-for-discrete-mathematics-paper

REFERENCES

[1] M. M. Solomon, "Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints," Operations Research, vol. 35, no. 2, pp. 254-265, 1987.

[2] P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," IEEE Transactions on Systems Science and Cybernetics, vol. 4, no. 2, pp. 100-107, 1968.

[3] J. R. Quinlan, "Induction of Decision Trees," Machine Learning, vol. 1, no. 1, pp. 81-106, 1986.

[4] J. A. Bondy and U. S. R. Murty, Graph Theory with Applications. London, UK: Macmillan Press, 1976.

[5] J. K. Lenstra and A. H. G. Rinnooy Kan, "Complexity of Vehicle Routing and Scheduling Problems," Networks, vol. 11, no. 2, pp. 221-227, 1981.

[6] R. Munir, Graf (Bagian 1), Program Studi Teknik Informatika, STEI-ITB. Available online: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2025-2026/20-Graf-Bagian1-2026.pdf [Accessed: June 14, 2026].

[7] R. Munir, Graf (Bagian 2), Program Studi Teknik Informatika, STEI-ITB. Available online: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2025-2026/21-Graf-Bagian2-2026.pdf [Accessed: June 14, 2026].

[8] R. Munir, Pohon (Bagian 1), Program Studi Teknik Informatika, STEI-ITB. Available online: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2025-2026/23-Pohon-Bag1-2026.pdf [Accessed: June 14, 2026].

[9] Google, "Google Maps Navigation and Geospatial Data," Available online: https://www.google.com/maps [Accessed: June 15, 2026].

[10] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.

[11] GraphicMaths, "Graph Theory Concepts and Types of Graphs," Available online: https://graphicmaths.com/computer-science/graph-theory/graphs/ [Accessed: June 17, 2026].

[12] DDS Wireless, "The Vehicle Routing Problem with Time Windows (VRPTW) in Fleet Management," Available online: https://ddswireless.com/scheduled-routes/ [Accessed: June 17, 2026].

[13] 365 Data Science, "Introduction to Decision Tree Classifiers," Available online: https://365datascience.com/tutorials/machine-learning-tutorials/decision-trees/ [Accessed: June 18, 2026].

[14] GeeksforGeeks, "A* Search Algorithm in AI and Graph Data Structures," Available online: https://www.geeksforgeeks.org/dsa/a-search-algorithm/ [Accessed: June 17, 2026].

STATEMENT

I hereby declare that the entirety of this paper is my original intellectual work. It has been composed independently without any form of plagiarism, unauthorized reproduction, or direct translation of previously published academic works.

Bandung, 19 June 2026

Bryan Pamungkas Prahara
13525092