Effective Memorizing of Bandung City Routes Using Graph Theory: Accelerating the Adaptation of Students Who Have Just Moved to Bandung

Marcel Luther Sitorus - 13524063[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]sitorusmarcel7@gmail.com, 13524063@std.stei.itb.ac.id

Abstract—This paper explores the use of graph theory to help new students at the Bandung Institute of Technology (ITB) adapt to Bandung City's complex urban routes. By modeling the city as a weighted graph, key graph algorithms such as Dijkstra's Algorithm, Minimum Spanning Tree (MST), and Closeness Centrality are used to find strategic locations, shortest paths, and optimal route plans. Using OpenStreetMap data and the OSMnx Python library, real-world geographic features are translated into graph representations, helping students visualize and memorize important city routes. The result is a data-driven model that enhances spatial understanding and supports student adaptation through intelligent navigation strategies.

I. Introduction

At Bandung Institute of Technology, after studying for one year in Jatinangor Town, the students will move to an appropriate location based on their study program. In this case, the collaborators will adapt to their new environment. One of the places that college students moved to is Bandung City. Bandung City is the third-largest City in Indonesia. With a large area of Bandung and a complex route, this can make college students feel overwhelmed.

This paper implements graph theory to find the most efficient way to adapt to Bandung City. This includes knowing where the most strategic boarding house is, the most efficient route to do college activities, and the most important routes to remember based on graph theory.

II. Graph Theory

A graph is a representation of a discrete structure that contains nodes(vertices) and edges. A graph can be noted as G = (V, E), where V is not non-empty set of nodes, and E is a set of edges that connect pairs of nodes (V). This also makes the graph represent the connection between nodes.

A. Type of Graph

There are so many types of graphs. Graphs can be classified as follows.

A.1. Simple Graph and Unsimple-Graph

Based on the presence or absence of double edges and ring edges in a graph, graphs can be grouped into two types as follows[1].

1. A Simple Graph is a graph that does not contain double edges and ring edges.

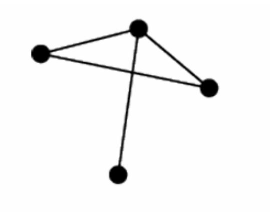

Figure 1 Illustration of Simple Graph
Source : Wolfram

2. *Unsimple-Graph*
   Unsimple-graph is a graph that contains double edges or ring edges, or both of edges.

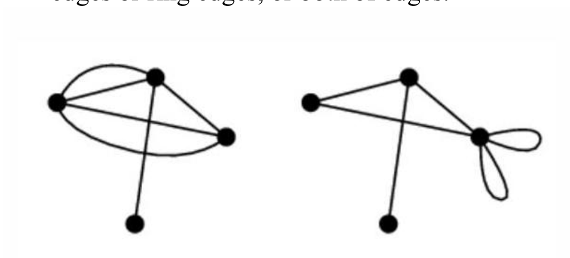

Figure 2 Illustration of Unsimple-Graph
Source: Wolfram

A.2. Undirected Graph and Directed Graph

Based on the direction orientation in the graph, there are two types of graphs as follows.

1. Undirected Graphs
   A graph with no direction orientation.

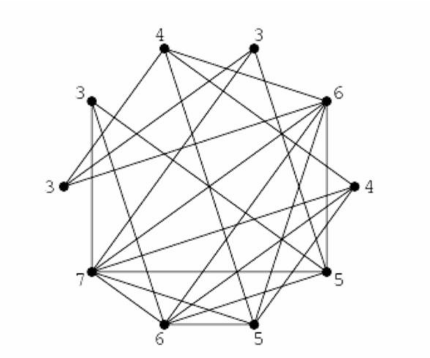

Figure 3 Illustration of Undirected Graph
Source : [1]

2. Directed Graph
   A graph with direction orientation.

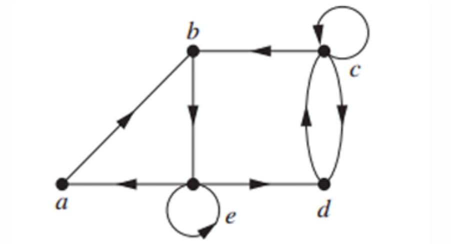

Figure 4 Illustration of Directed Graph
Source : [1]

Weight-Directed Graph

A weight-directed graph is a graph that contains the weights in every edge. The weight can represent distance or difficulties while going through the edges. That combines with the weight combines with directed graph.

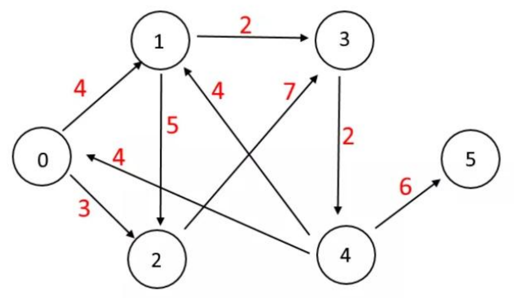

Figure 5 Illustration of Weight-Directed Graph
Source : [2]

B. Graph Terminology

Some terminologies are used in this paper. The first is adjacent. Two nodes are adjacent if there is at least one edge that connects the first node to the second node. The Second is Incidency. For arbitrary edges, $e = (v_j, v_k)$ implies e is incident on v_j node or v_k node. The third is a degree. Every node in a graph has a degree that is based on how many edges connect to the node. So if the nodes don't have any edges, the degree of the node is 0. And the fourth is path. A path of length n from a starting node v_0 to a goal node v_n in a graph G is an alternating sequence of vertices and edges of the form $v_0, e_1, v_1, e_2, v_2, \ldots, v_{n-1}, e_n, v_n$ such that $e_1 = (v_0, v_1), e_2 = (v_1, v_2), \ldots, e_n = (v_{n-1}, v_n)$ are the edges of the graph G. The fifth is cycle. Cycle is a path that started and ended in the same node.

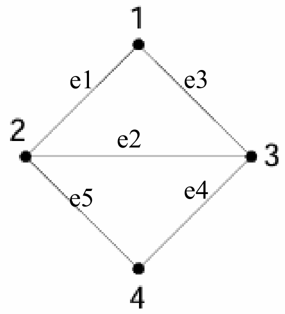

Figure 6 1 is adjacent to 2 and 3, e1 is incident on 1 and 2, the degree of 1 is 2, and e1, e3 are part of a path from 2 to 3
Source : [1]

III. Hamilton Circuit and Path

A Hamiltonian circuit (or Hamiltonian cycle) refers to a closed loop on a graph that visits each vertex exactly once and returns to the starting point. Formally, given a graph ($G = (V, E)$), a Hamiltonian circuit is a sequence of vertices ($v_1, v_2, \ldots, v_n, v_{n+1}$) such that:

1. Every vertex ($v_i \in V$) appears exactly once in the sequence (except the start/end vertex),
2. Each consecutive pair ($(v_i, v_{i+1}) \in E$),
3. And ($(v_n, v_1) \in E$) to complete the cycle

Hamilton circuits have the same rules as Euler circuits, the difference is that Hamilton circuits do not form circuits.

IV. Shortest Path Algorithm

To determine optimal travel routes, we apply shortest path algorithms such as Dijkstra's Algorithm. This algorithm finds the minimum total distance or travel time between two nodes.

Dijkstra's Algorithm is one of the most well-known algorithms for solving the single-source shortest path problem in a weighted graph with non-negative edge weights.

The algorithm maintains a set of vertices whose minimum distance from the source is known and repeatedly selects the vertex with the minimum tentative distance, updating the distances to its adjacent vertices if a shorter path is found through the selected vertex.

IV. Implementation of Graph Theory in Bandung City Routes

This study applies a quantitative and computational methodology grounded in discrete mathematics, particularly graph theory, to model and analyze the road network of Bandung City. The methodology consists of five main stages: data collection, graph modeling, algorithmic implementation, metric analysis, and visualization.

A. *Data Collection*

The primary data source for this study is OpenStreetMap (OSM), an open and editable geographic database. The following data elements were collected using the osmnx Python library[3]:

1. Street network data in the Bandung area (with a focus on walking and driving networks).

2. Point of Interest (POI) data, such as:
    a. Campus locations (e.g., ITB)
    b. Boarding houses (kos-kosan)
    c. Mosques, churches
    d. Cafes and printing stations
    e. Bookstores and other student-relevant location.

    Each POI includes spatial coordinates (latitude, longitude) and category tags.

B. Graph Modeling

The city road network is modeled as a weighted directed graph G=(V,E)G = (V, E)G=(V,E), where:

1. V: vertices representing geographic nodes (e.g., intersections, POIs, kos locations).
2. E: edges representing roads, with associated weights.

Two types of graphs are constructed:

1. Unweighted graphs, for theoretical problems like Hamiltonian paths and MST.
2. Weighted graphs, where edge weights are based on:
    a. Physical distance (in meters) from OSM.
    b. Estimated travel time (optional, based on assumed speeds or traffic data).

Graph simplification and projection are applied to ensure computational tractability and geographic consistency.

V. Analysis and Discussion

After the Bandung road network was successfully modeled as a graph using OpenStreetMap data and the osmnx Python library, several experiments were carried out to apply theoretical graph algorithms to practical navigation and memorization tasks.

A. Closeness Centrality Analysis

To evaluate which boarding houses (kos-kosan) offer the most strategic access to key locations (e.g., ITB, mosque, printing station, gym, bookstore), the closeness centrality of each kos node was computed. The centrality values were normalized and ranked, showing that boarding houses near Jalan Ganesha and Jalan Tamansari tend to have higher accessibility to multiple POIs. This provides quantitative insight into which kos-kosan are optimal for minimizing daily travel time.

B. Shortest Path Computation

Using Dijkstra's Algorithm, shortest paths were computed from ITB campus to various important destinations, such as:
- The nearest mosque (for daily worship)
- The nearest printing station (for assignment submission)
- Local cafes or bookstores (as potential study spots)

The resulting paths were visualized on a map with geographic accuracy, showing real-world roads and estimated travel distances. The algorithm's performance verified the theoretical prediction that Dijkstra can provide optimal pathfinding in real city networks, with all edge weights representing physical distance in meters.

C. Minimum Spanning Tree (MST)

An MST was constructed from a subgraph containing nodes representing only student-relevant POIs. This MST provides an ideal route for orientation or city exploration — a minimal-cost path that connects all key places without redundancy. The MST-based route can be suggested as a memorization tool for new students to cover all vital locations efficiently during their first week in Bandung.

D. Hamiltonian Path Approximation

Although finding an exact Hamiltonian path is NP-complete, a heuristic was used to simulate a route that visits each key location once. This simulated path mimics a self-guided orientation tour and may be recommended to students as a way to learn the city structure systematically while avoiding repeated locations. This supports the cognitive process of spatial memorization.

E. Visualization Output

The graph visualizations generated by matplotlib and contextily helped illustrate not only the structure of the urban network but also the decision-making process supported by graph metrics. Nodes were color-coded by centrality scores, while edges varied in width based on their travel weights. POIs were clearly overlaid on the city map to reinforce geographic context.

F. Discussion of Practicality

The analysis demonstrates that theoretical graph concepts can provide practical benefits in real-world adaptation scenarios. By combining mathematical modeling with real geographic data, students are empowered with data-driven tools for navigation, planning, and spatial understanding. While the current study assumes static traffic and edge weights, future iterations may incorporate real-time data for dynamic path planning.

VI. Conclusion

This paper demonstrates that graph theory offers a powerful framework for understanding and optimizing navigation in urban environments like Bandung. By modeling Bandung's city map as a graph and applying core algorithms such as Dijkstra's Algorithm, Minimum Spanning Tree, and Closeness Centrality, the study provides a clear strategy for route memorization and daily navigation tasks. New students can benefit from this system by gaining better spatial awareness and optimizing their movement between key places such as kos-kosan, campuses, and essential amenities. Further development may include integrating dynamic traffic data to improve accuracy in real-time scenarios.

A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

VII. APPENDIX

Python packages used in this study include:
- osmnx: for downloading and modeling street networks from OpenStreetMap
- networkx: for graph representation and algorithms
- matplotlib and geopandas: for visualizing graphs and geographic features
- contextily: for adding basemaps to visualizations

All simulations and visualizations were implemented using Python 3.10 in a Jupyter Notebook environment.

Appendixes, if needed, appear before the acknowledgment.

REFERENCES

[1] R. Munir, “Graf (Bag. 1),” Rinaldu Munir Webpage. Accessed: Jun. 20, 2025. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf

[2] H. Eren Keskin, “Directed Weighted Graph.” Accessed: Jun. 20, 2025. [Online]. Available: https://github.com/erenkeskin/directed-weighted-graph

[3] G. Boeing, “Modeling and Analyzing Urban Networks and Amenities With OSMnx,” *Geogr Anal*, May 2025, doi: 10.1111/gean.70009.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 21 Juni 2025

Marcel Luther Sitorus 13524063