Application of Graph Theory in Social Network Analysis among Himpunan Mahasiswa Informatika Institut Teknologi Bandung Class of 2023 on Instagram

Refki Alfarizi - 13523002[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]refkialfarizi46@gmail.com, 13523002@std.stei.ac.id

***Abstract*— This study applies graph theory to analyze the social network of Himpunan Mahasiswa Informatika (HMIF) Institut Teknologi Bandung's Class of 2023 on Instagram. By manually collecting data from Instagram follower lists, interactions and relationships among members were modeled as a graph. Various graph-theoretical metrics, including centrality and community detection using Louvain Method, were utilized to evaluate the network's structure and dynamics. The analysis identified tightly-knit communities and provided insights into the overall connectivity and information flow among members. This paper demonstrates the practical application of graph theory in social network analysis within an academic setting, serving as an assignment for a discrete mathematics course.**

***Keywords*—Graph Theory, Instagram, NetworkX, Social Network Analysis.**

I. INTRODUCTION

In today's digital landscape, social media platforms play a pivotal role in shaping and maintaining social networks, especially within academic communities. Instagram, a widely used social networking service, facilitates interactions among members through various forms of engagement such as follows, likes, and comments. Understanding the structure and dynamics of these interactions is essential for organizations aiming to enhance communication, foster community spirit, and optimize information dissemination. This study focuses on the Himpunan Mahasiswa Informatika (HMIF) at Institut Teknologi Bandung, specifically analyzing the Class of 2023's interactions on Instagram through the application of graph theory.

Graph theory, a fundamental branch of discrete mathematics, provides robust tools for modelling and analysing relationships within a network. By representing individuals as nodes and their interactions as edges, graph-theoretical metrics can elucidate the underlying structure of the network, identify key influencers, and detect community formations. These metrics include centrality measures, which highlight the most influential members, density calculations that assess the overall connectivity, and clustering coefficients that reveal the presence of tightly knit groups within the network. The utilization of these metrics allows for a comprehensive understanding of the social dynamics at play.

This research focuses on constructing a social graph based on manually collected data from Instagram follower lists of HMIF's Class of 2023. The analysis aims to quantify and interpret the network's characteristics, identifying central members who drive engagement, assessing the network's connectivity, and uncovering subgroups with high interaction levels. By applying graph-theoretical concepts learned in the discrete mathematics course, this study provides practical insights into the application of theoretical models in real-world social networks.

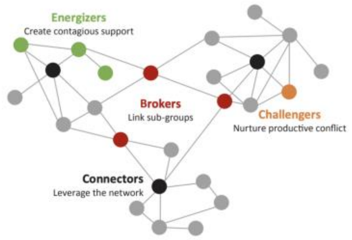

Figure 1.1 Illustration of Social Network Analysis
https://iriss.colostate.edu/nest/social-network-analysis/

II. THEORETICAL BASIS

A. Graph

Graphs are discrete structures made up of vertices and edges that link these vertices. A graph G is an ordered pair (V, E) where V is a non-empty set of vertices (also called nodes) and $E \subseteq V \times V$ is a set of edges [1]. Each edge $\{u, v\} \in E$ (or $\langle u, v\rangle$ for directed edges) denotes a connection between vertices u and v. When dealing with *undirected* graphs, the notation $\{u, v\}$ implies that edges do not carry a direction; conversely, for directed graphs, the notation $\langle u, v\rangle$ implies an orientation from u to v.

$$G = (V, E), V \neq \emptyset, E \subseteq V \times V.$$

Let $n = |V|$ denote the number of vertices and $m = |E|$ denote the number of edges.

$$n = |V|, m = |E|.$$

1. Simple Graph—A graph with no loops (edges of the form $\{v, v\}$) and no multiple edges between the same pair of vertices.

2. Multigraph— A graph that may contain multiple edges (parallel edges) between two vertices and possibly loops.

3. Weighted Graph—A graph where each edge (u, v) has an associated real number (or weight) $w(u, v)$.

4. Directed Graph—A graph where each edge has an orientation, denoted $\langle u, v\rangle$, which differs from $\langle v, u\rangle$.

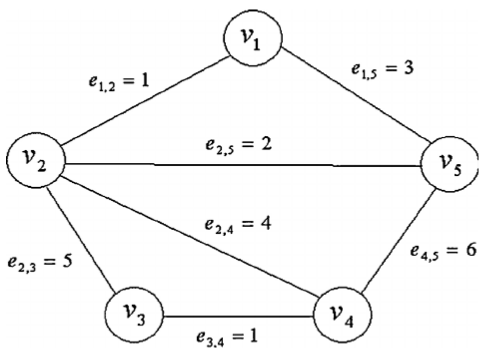

Figure 2.1
Example illustration of an undirected graph G=(V,E)
https://www.researchgate.net/figure/An-undirected-graph-G-with-5-vertex-and-7-edges_fig1_250922991

Here are some key properties of graph.

1. Degree
- Undirected case. The degree of a vertex v, denoted $deg(v)$, is the number of edges incident to v. Formally,

$$deg(v) = |\{u | \{u, v\} \in E\}|.$$

- Directed case. We distinguish in-degree $deg^{-}(v)$ and out-degree $deg^{+(v)}$.

$$deg^{-}(v) = |\{u | \langle u, v\rangle \in E\}|, deg^{+\cdot}(v) = |\{w | \langle v, w\rangle \in E\}|$$

2. Path and Connectivity
A path P in G from v_1 to v_k is a sequence of vertices $(v_1, v_2, \ldots, v_k)$ such that $\{v_i, v_{i+1}\} \in E$ for all $1 \leq i < k$. A graph is connected if for any two vertices u and v, there exists a path between them.

3. Subgraph
A subgraph $H = (V_H, E_H)$ of $G = (V, E)$ is a graph where $V_H \subseteq V$ and $E_H \subseteq E$ such that any edge in E_H connects only vertices in V_H.

4. Bipartite Graph
A graph (V, E) is bipartite if its vertex set V can be decomposed into two disjoint sets V_1 and V_2 such that no edge has both endpoints within V_1 or within V_2. Formally,

$$E \subseteq (V_1 \times V_2) \cup (V_2 \times V_1), V = V_1 \cup V_2, V_1 \cap V_2 = \emptyset.$$

These discrete properties form the backbone of analyzing complex systems modeled by graphs, including social networks, communication systems, and other relational structures.

B. Social Network Analysis

Social Network Analysis (SNA) leverages graph theory to study social structures formed by individuals (or entities) and their interactions [2]. In this framework:

- Vertices often represent people, organizations, or other social units.

- Edges represent relationships or interactions (e.g., friendships, communication links).

C. Graph Representation of Social Networks

Computational tasks often require an explicit data structure to store and manipulate graphs. Two principal representations are the adjacency matrix and the adjacency list.

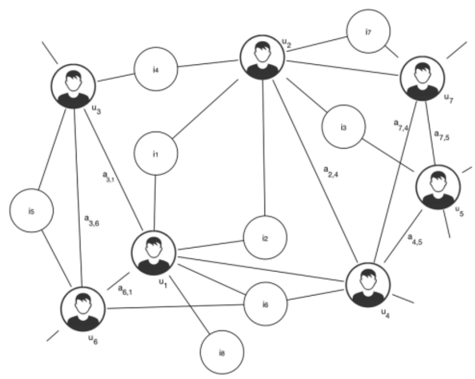

Figure 2.2
Example of a small social network represented as a graph
https://www.researchgate.net/figure/A-graph-representation-of-a-social-network_fig1_318357963

1. Adjacency List
The adjacency list representation is often preferred for large or sparse social networks. Formally, an adjacency list can be defined as a function:

$$N: V \rightarrow P(V),$$

where $P(V)$ denotes the power set of V. For every vertex $v \in V$, $N(v)$ is the set of vertices adjacent to v. Hence, for an undirected graph,

$$u \in N(v) \Leftrightarrow \{v, u\} \in E.$$

For a directed graph,

$$u \in N(v) \Leftrightarrow \langle v, u \rangle \in E.$$

Advantages:
- Space Efficiency: For sparse graphs (where $m \ll n^2$), adjacency lists store only existing edges.
- Neighborhood Operations: Quickly enumerate neighbors of any vertex v.

Drawbacks:
- Less direct for certain matrix-based operations (e.g., matrix multiplication, spectral analysis).

2. Adjacency Matrix

An adjacency matrix A is an $n \times n$ matrix (with $n = |V|$) defined by:

$$A_{ij} = \begin{Bmatrix} 1, & \text{if}\{v_i, v_j\} \in E(\text{unweighted case}), \\ w(v_i, v_j), & \text{if}\langle v_i, v_j \rangle \in E(\text{weighted/directed case}), \\ 0, & \text{otherwise.} \end{Bmatrix}$$

Adjacency matrices are favoured when performing algebraic operations such as eigenvalue/eigenvector computations or matrix decompositions for community detection.

D. Louvain Algorithm for Community Detection

Once a social network is represented in a computationally accessible form (e.g., via adjacency lists), the Louvain algorithm is a popular method for detecting communities based on modularity optimization [3].

1. Modularity Q

Louvain aims to maximize a modularity function defined for an undirected, weighted network as:

$$Q = \frac{1}{2m} \sum_{i,j} \left(a_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j),$$

Where.

- m is the total weight of edges in the graph (for an unweighted graph, m is simply the number of edges),
- $k_i = \sum_j a_{ij}$ is the weighted degree of vertex i,
- c_i is the community assignment of vertex i, and $\delta(c_i, c_j)$ is 1 if $c_i = c_j$ and 0 otherwise [4].

2. Two-Phase Iterative Process

- Local moving: Initialize each vertex in its own community. Iteratively move individual vertices to neighboring communities if it yields a higher Q.
- Aggregation: Merge each discovered community into a single “super-vertex” to form a new network. Edges between super-vertices are weighted by the sum of edges between vertices in the original communities.

Repeat these two phases until no improvement in Q is possible. This hierarchical approach scales well to large networks, making it highly suitable for SNA.

E. Centrality Measures

In SNA, centrality measures quantify the “importance” or “influence” of vertices. Three widely used measures include:

1. Degree Centrality

For an undirected network, the degree centrality of vertex v is:

$$C_D(v) = deg(v).$$

Vertices with larger $deg(v)$ have more direct ties and may be considered more visible or influential.

2. Betweenness Centrality

Betweenness centrality gauges the extent to which a vertex bridges different parts of the network. Formally:

$$C_B(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}},$$

where σ_{st} is the total number of shortest paths from s to t and $\sigma_{st}(v)$ is the number of those paths passing through v.

3. Closeness Centrality

Closeness centrality measures the reciprocal of the average distance from v to every other vertex:

$$C_C(v) = \frac{1}{\sum_{u \in V} d\,(v, u)},$$

where $d(v, u)$ denotes the length of the shortest path between v and u.

III. METHODOLOGY

A. Data Collection

The initial phase of this study involves the meticulous manual collection of relevant data from Instagram, tailored to the specific context of the HMIF Class of 2023. Given the manageable size of the dataset, manual collection ensures accuracy and completeness, circumventing the potential limitations and ethical concerns associated with automated data scraping. Two primary data sources are utilized:

1. User Metadata

This dataset includes critical attributes for each Instagram user within the HMIF Class of 2023, specifically username and private account status.

2. Connection Data

This dataset captures the follower-followee relationships among Instagram users, representing the edges of the social network graph. Each record delineates a connection between two users, effectively mapping the relational structure of the network.

Manual data collection was chosen to maintain data integrity and ensure adherence to Instagram's terms of service. This approach, while labour-intensive, guarantees that the dataset authentically reflects the genuine interactions and relationships within the specified academic cohort.

B. Data Preprocessing

Ensuring data quality and consistency is paramount for the reliability of subsequent analyses. The preprocessing phase encompasses several steps.

1. Data Cleaning

The collected datasets undergo rigorous cleaning procedures to eliminate inconsistencies such as duplicate entries and incomplete records. Duplicate connections, which may arise from bidirectional relationships (e.g., User A follows User B and User B follows User A), are identified and consolidated to maintain a streamlined dataset.

2. Data Integration

The user metadata and connection data are integrated to form a cohesive dataset. This integration ensures that all connections reference valid and existing users, thereby preserving the integrity of the social network representation.

3. Handling Missing Data

Instances where connections reference users absent from the metadata are addressed by excluding such connections. This exclusion prevents distortions in the network analysis and ensures that all nodes in the graph are accounted for with corresponding metadata.

4. Data Transformation

The cleaned and integrated data is transformed into a suitable format for graph construction. This involves structuring the data to effectively represent nodes (users) and edges (connections), facilitating a seamless transition into graph-based modeling.

C. Graph Construction Using NetworkX

The construction of the social network graph is undertaken using the NetworkX library, a robust tool for graph creation and analysis in Python. NetworkX offers comprehensive functionalities that align with the requirements of this study, enabling the effective modeling of complex social networks.

1. NetworkX

NetworkX is an open-source Python library designed for the creation, manipulation, and study of complex networks. It provides an extensive suite of graph-theoretical algorithms and data structures that are essential for social network analysis. NetworkX supports various types of graphs, including undirected, directed, and weighted graphs, making it highly versatile for modelling diverse network structures.

2. Graph Construction Procedure

Connections between users are incorporated into the graph as edges based on the cleaned connection data. Each edge signifies a follower-followee relationship, encapsulating the social interactions that define the network's structure. To maintain the validity of the social network representation, self-loops—edges connecting a node to itself—are identified and removed, as they do not contribute meaningful information in this context and can skew analytical metrics.

NetworkX's seamless handling of nodes and edges, coupled with its support for various graph-theoretical algorithms, facilitates the efficient construction and manipulation of the social network graph. This foundational graph serves as the basis for subsequent exploratory.

D. Exploratory Data Analysis (EDA)

This phase involves the systematic examination of the network's structural properties to inform subsequent analytical procedures. The key components of EDA in this study include:

1. Degree Distribution Analysis

Assessing the distribution of connections among users to identify patterns of connectivity and the presence of highly connected nodes or hubs within the network.

2. Identification of Influential Users

Pinpointing users with the highest number of connections to understand their potential role in information dissemination and network cohesion.

3. Privacy Status Correlation

Analysing the relationship between users' privacy settings and their network positions to determine how privacy preferences influence connectivity and interaction patterns.

E. Graph Visualization

Visualizing the social network graph is essential for intuitively understanding its complex structure and identifying key patterns. This phase employs advanced visualization techniques to represent the network's topology while maintaining user privacy through anonymization. The visualization strategy encompasses:

1. Anonymization of User Data

To safeguard user privacy, actual usernames are replaced with unique identifiers (e.g., Node 1, Node 2). This ensures that visual representations do not disclose personal information.

2. Layout Algorithms

Utilizing layout algorithms such as the spring layout to position nodes in a manner that reflects their connections, facilitating the identification of clusters and central nodes.

3. Color-Coding Based on Attributes

Implementing colour schemes to differentiate between private and public accounts, enabling a clear visual distinction of privacy dynamics within the network.

4. Color-Coding Based on Attributes

Displaying detected communities through distinct color palettes or boundaries to illustrate subgroups within the network, revealing clusters of users with dense interconnections.

F. Implementation of Centrality Measures

Centrality measures are pivotal in quantifying the importance and influence of individual nodes within the social network. This study implements several centrality metrics using NetworkX to evaluate and interpret the network's dynamics.

1. Degree Centrality

Calculates the number of direct connections each node possesses, highlighting users with high connectivity and potential influence within the network.

2. Betweenness Centrality

Measures the extent to which a node lies on the shortest paths between other nodes, identifying users that act as bridges or gatekeepers facilitating interactions across different network segments.

3. Closeness Centrality

Assesses how close a node is to all other nodes in the network, reflecting its ability to interact with the entire network efficiently and disseminate information swiftly.

F. Ethical Considerations

Ethical considerations are integral to the methodological framework of this study, ensuring that the analysis respects user privacy and adheres to relevant guidelines. The primary ethical protocols implemented include:

1. Data Anonymization

All user data is anonymized by replacing actual usernames with unique identifiers, preventing the disclosure of personal information in visualizations and analyses.

2. Compliance with Terms of Service

The data collection and analysis processes strictly adhere to Instagram's terms of service, ensuring that the study does not infringe upon platform policies or user rights.

3. Confidentiality Assurance

Measures are in place to protect the confidentiality of the collected data, ensuring that sensitive information is not exposed or misused during the analysis.

IV. Algorithm and Implementation

This section elucidates the algorithms employed and their implementation within the analysis of the Himpunan Mahasiswa Informatika (HMIF) Institut Teknologi Bandung's Class of 2023 Instagram network. The implementation is executed in Python, utilizing the NetworkX library for graph construction and analysis, alongside essential libraries such as pandas, matplotlib, and seaborn. Each component is detailed through corresponding code snippets and comprehensive explanations to facilitate a clear understanding of the operational workflow.

A. Graph Construction Using NetworkX

The construction of the social network graph is foundational to the analysis, enabling the application of graph-theoretical metrics and visualization techniques. NetworkX, a versatile Python library, is leveraged to create and manipulate the graph structure efficiently.

Initially, an undirected graph G is initialized to represent the mutual follower-followee relationships among Instagram users. Each user from the metadata is added as a node with an associated attribute indicating their privacy status. Subsequently, connections between users are established as edges based on the pre-processed connection data. To ensure the integrity of the graph, self-loops—edges where a node connects to itself—are identified and removed, as they do not contribute meaningful information to the analysis.

```
import networkx as nx

G = nx.Graph()

for _, row in df_metadata.iterrows():
 G.add_node(row['username'],
is_private=row['is_private'])

for _, row in df_connections_unique.iterrows():
 G.add_edge(row['node_1'], row['node_2'])

self_loops = list(nx.selfloops(G))
if self_loops:
 G.remove_edges_from(self_loops)
```

The above code initializes an undirected graph and populates it with nodes and edges derived from the metadata and connection data. By iterating through each row of the metadata DataFrame, nodes are added with the is_private attribute. Edges are then formed by iterating through the unique connections DataFrame, establishing relationships between users. The final step ensures the removal of any self-loops to maintain the graph's validity.

B. Exploratory Data Analysis (EDA)

Exploratory Data Analysis serves as a preliminary step to uncover initial patterns and insights within the social network. This phase involves calculating and visualizing key graph metrics to inform deeper analytical procedures.

1. Degree Distribution Analysis

The degree distribution provides an overview of the connectivity patterns within the network, highlighting the number of connections each user has. Analysing this distribution helps identify highly connected users, often referred to as hubs, which play significant roles in information dissemination and network cohesion.

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

degree_dict = dict(G.degree())
df_degree =
pd.DataFrame(list(degree_dict.items()),
columns=['username', 'degree'])

plt.figure(figsize=(10,6))
sns.histplot(df_degree['degree'],
bins=range(df_degree['degree'].min(),
df_degree['degree'].max() + 2), kde=False,
color='skyblue', edgecolor='black')
plt.title('Degree Distribution')
plt.xlabel('Degree (Number of Connections)')
plt.ylabel('Number of Users')
plt.xticks(range(df_degree['degree'].min(),
df_degree['degree'].max() + 1))
plt.show()
```

This code calculates the degree of each node in the graph and visualizes the distribution using a histogram. The plot elucidates how connections are spread across users, identifying common degrees and the presence of highly connected individuals.

2. Identification of Top Connected Users

Identifying users with the highest degrees is essential for pinpointing influential figures within the network. These users can significantly impact information flow and network dynamics.

```
top_degree = df_degree.sort_values(by='degree',
ascending=False).head(10)
print("Top 10 Users by Degree:")
print(top_degree)
```

The above snippet sorts the users based on their degree and extracts the top ten most connected users. These users are pivotal in maintaining network connectivity and facilitating interactions.

3. Private Account Distribution

Analyzing the distribution of private versus public accounts offers insights into how privacy preferences influence network interactions and overall connectivity.

```
privacy_counts =
df_metadata['is_private'].value_counts().rename(
index={False: 'Public', True: 'Private'})
print("Privacy Status Counts:")
print(privacy_counts)
```

This code counts the number of private and public accounts within the network, providing a clear understanding of privacy dynamics and their prevalence among users.

C. Graph Visualization

Visual representations are crucial for intuitively understanding the network's structure and identifying key patterns. This phase employs advanced visualization techniques while ensuring user privacy through anonymization.

1. Anonymization of User Data

To protect user privacy, actual usernames are anonymized by assigning unique identifiers. This step ensures that visualizations do not disclose personal information.

```
username_list = list(G.nodes())
anonymized_ids = [f'Node {i+1}' for i in
range(len(username_list))]
mapping = dict(zip(username_list,
anonymized_ids))
G_anonymized = nx.relabel_nodes(G, mapping)
```

The code replaces actual usernames with anonymized identifiers, creating a new graph G_anonymized that maintains the original structure without revealing personal information.

2. Basic Network Plotting

A fundamental visualization of the anonymized graph is created using the spring layout algorithm, which positions nodes based on their connections to provide an aesthetically pleasing and informative layout.

```
plt.figure(figsize=(20, 20))
pos = nx.spring_layout(G_anonymized, k=0.15,
iterations=20)
nx.draw_networkx_nodes(G_anonymized, pos,
node_size=300, node_color='skyblue', alpha=0.7)
nx.draw_networkx_edges(G_anonymized, pos,
width=1.0, alpha=0.5)
plt.title('Social Network Graph of ITB
Informatics Class of 2023 on Instagram',
fontsize=20)
plt.axis('off')
plt.show()
```

This visualization provides an overarching view of the network's structure, highlighting clusters and isolated nodes without revealing user identities.

3. Color-Coding Based on Privacy Status

Nodes are color-coded to differentiate between private and public accounts, enabling a visual assessment of privacy dynamics within the network.

```
node_colors = ['red' if
G_anonymized.nodes[node]['is_private'] else
'green' for node in G_anonymized.nodes()]

plt.figure(figsize=(20, 20))
pos = nx.spring_layout(G_anonymized, k=0.15,
iterations=20)
nx.draw_networkx_nodes(G_anonymized, pos,
node_size=300, node_color=node_colors,
alpha=0.7)
nx.draw_networkx_edges(G_anonymized, pos,
width=1.0, alpha=0.3)
plt.title('Anonymized Social Network Graph
Highlighting Private (Red) and Public (Green)
Accounts', fontsize=20)
plt.axis('off')

import matplotlib.patches as mpatches
```

```
private_patch = mpatches.Patch(color='red',
label='Private Account')
public_patch = mpatches.Patch(color='green',
label='Public Account')
plt.legend(handles=[private_patch,
public_patch], loc='best')
plt.show()
```

The resulting plot visually distinguishes private and public accounts, facilitating the analysis of how privacy settings correlate with network connectivity and interactions.

4. Community Visualization

In this visualization, we employ the Louvain algorithm, a popular and efficient method for community detection in large networks. The algorithm iteratively optimizes modularity—a measure of the strength of division of a network into communities—making it particularly well-suited for detecting well-defined groups within the anonymized graph.

Using the community_louvain library, the graph's nodes are partitioned into distinct communities based on their interconnectivity. This process results in the detection of num_communities distinct communities, each represented by a unique color in the visualization. The Louvain algorithm's effectiveness lies in its ability to detect these communities while maintaining computational efficiency, even for complex, large-scale networks.

```
import community as community_louvain

partition =
community_louvain.best_partition(G_anonymized)
num_communities = len(set(partition.values()))
cmap = sns.color_palette("hsv", num_communities)
community_colors = {node: cmap[comm] for node,
comm in partition.items()}
colors = [community_colors.get(node, 'black')
for node in G_anonymized.nodes()]

plt.figure(figsize=(20, 20))
pos = nx.spring_layout(G_anonymized, k=0.15,
iterations=20)
nx.draw_networkx_nodes(G_anonymized, pos,
node_size=300, node_color=colors, alpha=0.7)
nx.draw_networkx_edges(G_anonymized, pos,
width=1.0, alpha=0.3)
plt.title('Anonymized Social Network Graph with
Detected Communities (Louvain Method)',
fontsize=20)
plt.axis('off')
plt.show()

from matplotlib.lines import Line2D

legend_elements = [Line2D([0], [0], marker='o',
color='w', label=f'Community {i+1}',

markerfacecolor=cmap[i], markersize=10) for i in
range(num_communities)]
plt.legend(handles=legend_elements, loc='best')
plt.show()
```

This visualization highlights distinct communities within the network, providing insights into the modular structure and the presence of closely-knit user groups.

D. Implementation of Centrality Measures

Centrality measures are pivotal in quantifying the importance and influence of individual nodes within the social network. Utilizing NetworkX's graph-theoretical capabilities, several centrality metrics are implemented to identify key influencers and understand the network's dynamics.

1. Degree Centrality

Degree centrality quantifies the number of direct connections each node possesses, highlighting users with high connectivity and potential influence within the network.

```
degree_centrality_anonymized =
nx.degree_centrality(G_anonymized)
df_degree_centrality_anonymized =
pd.DataFrame(list(degree_centrality_anonymized.i
tems()), columns=['Node', 'Degree Centrality'])
df_degree_centrality_anonymized =
df_degree_centrality_anonymized.sort_values(by='
Degree Centrality', ascending=False)
print("\nTop 5 Nodes by Degree Centrality:")
print(df_degree_centrality_anonymized.head(5))
```

This code computes the degree centrality for each node, sorts them in descending order, and identifies the top five nodes with the highest centrality scores, indicating their prominence within the network.

2. Betweenness Centrality

Betweenness centrality measures the extent to which a node lies on the shortest paths between other nodes, identifying users that act as bridges or gatekeepers facilitating interactions across different network segments.

```
betweenness_centrality_anonymized =
nx.betweenness_centrality(G_anonymized)
df_betweenness_anonymized =
pd.DataFrame(list(betweenness_centrality_anonymi
zed.items()), columns=['Node', 'Betweenness
Centrality'])
df_betweenness_anonymized =
df_betweenness_anonymized.sort_values(by='Betwee
nness Centrality', ascending=False)
print("\nTop 5 Nodes by Betweenness
Centrality:")
print(df_betweenness_anonymized.head(5))
```

By calculating betweenness centrality, this snippet identifies nodes that play crucial roles in connecting different parts of the network, thereby facilitating information flow and maintaining network cohesion.

3. Closeness Centrality

Closeness centrality assesses how close a node is to all other nodes in the network, reflecting its ability to interact with the entire network efficiently and disseminate information swiftly.

```
closeness_centrality_anonymized =
nx.closeness_centrality(G_anonymized)
df_closeness_anonymized =
pd.DataFrame(list(closeness_centrality_anonymize
d.items()), columns=['Node', 'Closeness
Centrality'])
df_closeness_anonymized =
df_closeness_anonymized.sort_values(by='Closenes
s Centrality', ascending=False)
print("\nTop 5 Nodes by Closeness Centrality:")
print(df_closeness_anonymized.head(5))
```

This code calculates the closeness centrality for each node, sorting them to reveal users who can efficiently interact with the entire network, thereby playing significant roles in rapid information dissemination.

V. Experiment Result

This section presents the empirical findings derived from the application of graph-theoretical metrics to the constructed social network of the Himpunan Mahasiswa Informatika (HMIF) Institut Teknologi Bandung's Class of 2023 on Instagram. The analysis encompasses centrality measures, community detection, network density, and average path length, providing a comprehensive understanding of the network's structural properties and user dynamics within the student community.

A. Degree Centrality

Degree centrality quantifies the number of direct connections each node possesses within the network, serving as a primary indicator of a user's influence and connectivity. The top five nodes with the highest degree centrality are as follows.

```
Top 5 Nodes by Degree Centrality:
```

	Node	Degree Centrality
138	Node 139	0.854478
165	Node 166	0.787313
148	Node 149	0.738806
3	Node 4	0.652985
25	Node 26	0.645522

Figure 5.1 Top 5 Nodes by Degree Centrality

Node 139 emerges as the most influential user with a degree centrality of 0.854478, indicating a high number of direct connections within the network. This suggests that Node 139 is a central hub, likely an active participant or leader within the HMIF Class of 2023. Similarly, Nodes 166 and 149 also exhibit substantial degree centrality scores, positioning them as key influencers who can effectively disseminate information and foster interactions among peers. Nodes 4 and 26, while slightly lower, still maintain significant connectivity, underscoring their roles in maintaining network cohesion.

B. Betweenness Centrality

Betweenness centrality measures the extent to which a node lies on the shortest paths between other nodes, highlighting its role as a bridge or intermediary within the network. The top five nodes with the highest betweenness centrality are as follows.

```
Top 5 Nodes by Betweenness Centrality:
```

	Node	Betweenness Centrality
138	Node 139	0.037661
165	Node 166	0.035603
148	Node 149	0.019881
96	Node 97	0.014244
3	Node 4	0.013986

Figure 5.2 Top 5 Nodes by Betweeness Centrality

Node 139 not only holds the highest degree centrality but also the highest betweenness centrality (0.037661), indicating its pivotal role in bridging different parts of the network. This dual centrality suggests that Node 139 is instrumental in facilitating interactions across various subgroups within the student community. Nodes 166 and 149 similarly serve as critical connectors, ensuring smooth information flow and collaboration between diverse clusters. Nodes 97 and 4, while less central, still play important intermediary roles, contributing to the network's overall connectivity.

C. Closeness Centrality

Closeness centrality assesses how close a node is to all other nodes in the network, reflecting its ability to interact efficiently with the entire network. The top five nodes with the highest closeness centrality are as follows.

```
Top 5 Nodes by Closeness Centrality:
```

	Node	Closeness Centrality
138	Node 139	0.868252
165	Node 166	0.818321
148	Node 149	0.785688
3	Node 4	0.733909
25	Node 26	0.729727

Figure 5.3 Top 5 Nodes by Closeness Centrality

Node 139 leads in closeness centrality (0.868252), indicating its strategic position for rapid dissemination of information across the network. Nodes 166 and 149 also exhibit high closeness centrality, suggesting their efficiency in reaching other users swiftly. High closeness centrality is indicative of nodes that can facilitate quick information spread and enhance overall network responsiveness. Leveraging these nodes can improve the network's ability to disseminate announcements, updates, and important information effectively.

D. Community Detection

Community detection identifies subgroups within the network where nodes are more densely connected internally than with the rest of the network. Utilizing the Louvain Method, the analysis detected ten distinct communities within the social network.

```
Detecting communities using the Louvain method...

Number of communities detected: 10
Communities:
Community 1: ['Node 1', 'Node 101', 'Node 102', 'Node 103', 'Node 108', 'Node 110', 'Node 112', 'Node 114', 'Node 116', 'Node 1
18', 'Node 119', 'Node 12', 'Node 125', 'Node 126', 'Node 129', 'Node 133', 'Node 134', 'Node 142', 'Node 148', 'Node 159', 'No
de 161', 'Node 163', 'Node 164', 'Node 165', 'Node 167', 'Node 170', 'Node 172', 'Node 173', 'Node 176', 'Node 177', 'Node 17
8', 'Node 18', 'Node 183', 'Node 187', 'Node 192', 'Node 195', 'Node 2', 'Node 209', 'Node 21', 'Node 219', 'Node 220', 'Node 2
21', 'Node 238', 'Node 247', 'Node 254', 'Node 255', 'Node 256', 'Node 257', 'Node 26', 'Node 260', 'Node 261', 'Node 264', 'No
de 266', 'Node 267', 'Node 29', 'Node 38', 'Node 4', 'Node 40', 'Node 46', 'Node 47', 'Node 48', 'Node 5', 'Node 54', 'Node 5
8', 'Node 62', 'Node 65', 'Node 66', 'Node 72', 'Node 75', 'Node 80', 'Node 83', 'Node 85', 'Node 90', 'Node 91', 'Node 93', 'N
ode 96', 'Node 98']
Community 2: ['Node 10', 'Node 100', 'Node 107', 'Node 111', 'Node 113', 'Node 117', 'Node 120', 'Node 121', 'Node 123', 'Node
124', 'Node 127', 'Node 128', 'Node 13', 'Node 130', 'Node 131', 'Node 135', 'Node 136', 'Node 139', 'Node 14', 'Node 140', 'No
de 141', 'Node 143', 'Node 145', 'Node 15', 'Node 150', 'Node 151', 'Node 155', 'Node 158', 'Node 16', 'Node 160', 'Node 166',
'Node 168', 'Node 17', 'Node 171', 'Node 179', 'Node 184', 'Node 19', 'Node 191', 'Node 194', 'Node 196', 'Node 197', 'Node 19
9', 'Node 200', 'Node 202', 'Node 203', 'Node 205', 'Node 206', 'Node 208', 'Node 210', 'Node 212', 'Node 213', 'Node 216', 'No
de 217', 'Node 218', 'Node 22', 'Node 222', 'Node 223', 'Node 227', 'Node 230', 'Node 231', 'Node 232', 'Node 234', 'Node 235',
'Node 236', 'Node 242', 'Node 243', 'Node 245', 'Node 250', 'Node 252', 'Node 253', 'Node 28', 'Node 3', 'Node 30', 'Node 31',
'Node 32', 'Node 35', 'Node 37', 'Node 43', 'Node 49', 'Node 51', 'Node 52', 'Node 55', 'Node 57', 'Node 59', 'Node 60', 'Node
63', 'Node 68', 'Node 69', 'Node 7', 'Node 71', 'Node 74', 'Node 78', 'Node 8', 'Node 84', 'Node 88', 'Node 92', 'Node 94', 'No
de 95', 'Node 99']
Community 3: ['Node 104', 'Node 105', 'Node 109', 'Node 11', 'Node 115', 'Node 122', 'Node 132', 'Node 138', 'Node 149', 'Node
152', 'Node 153', 'Node 157', 'Node 169', 'Node 174', 'Node 182', 'Node 188', 'Node 189', 'Node 193', 'Node 198', 'Node 201',
'Node 204', 'Node 224', 'Node 225', 'Node 228', 'Node 229', 'Node 23', 'Node 237', 'Node 239', 'Node 24', 'Node 240', 'Node 24
4', 'Node 246', 'Node 25', 'Node 269', 'Node 27', 'Node 33', 'Node 36', 'Node 39', 'Node 42', 'Node 44', 'Node 45', 'Node 50',
'Node 6', 'Node 61', 'Node 67', 'Node 70', 'Node 76', 'Node 77', 'Node 86', 'Node 87', 'Node 89', 'Node 97']
Community 4: ['Node 268']
Community 5: ['Node 106', 'Node 137', 'Node 144', 'Node 146', 'Node 154', 'Node 156', 'Node 162', 'Node 180', 'Node 181', 'Node
185', 'Node 186', 'Node 20', 'Node 207', 'Node 211', 'Node 214', 'Node 215', 'Node 226', 'Node 233', 'Node 248', 'Node 249', 'N
ode 251', 'Node 258', 'Node 259', 'Node 262', 'Node 263', 'Node 265', 'Node 34', 'Node 41', 'Node 53', 'Node 56', 'Node 64', 'N
ode 79', 'Node 81', 'Node 82', 'Node 9']
Community 6: ['Node 73']
Community 7: ['Node 147']
Community 8: ['Node 175']
Community 9: ['Node 190']
Community 10: ['Node 241']
```

Figure 5.4 Detected Communities

The detection of ten communities reveals the presence of tightly-knit subgroups within the network, each potentially representing clusters of users with shared interests, affiliations, or close personal connections. For instance, Community 1 comprises a significant number of nodes, indicating a robust subgroup with dense interconnections. Understanding these communities is crucial for targeted engagement strategies, as interventions can be tailored to the specific characteristics and dynamics of each subgroup. Additionally, the identification of isolated communities, such as Community 4 (Node 268) and Communities 6-10, highlights the existence of niche or less connected user groups that may require specialized approaches to foster integration and interaction within the broader network.

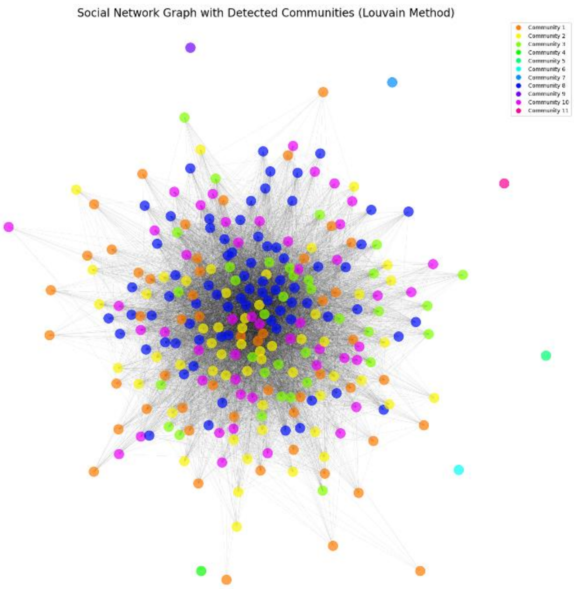

Figure 5.5 Detected Communities Visualized

E. Network Density

Network density measures the proportion of actual connections to possible connections within the network, providing insights into the overall connectivity and cohesion.

```
Network Density: 0.2631
```

Figure 5.6 Network Density Result

A network density of 0.2631 indicates that approximately 26.31% of all possible connections between users are present. This moderate density suggests a reasonably interconnected network, allowing for effective information flow while also presenting opportunities to enhance connectivity. The consistency in density values between the original and anonymized graphs confirms the integrity of the anonymization process. To further strengthen the network, strategies could be implemented to encourage additional interactions, thereby increasing density and fostering a more cohesive community.

F. Average Path Length

Average path length evaluates the typical number of steps required to traverse from one node to another within the network, reflecting the network's efficiency in information dissemination.

```
Average Path Lengths for Connected Components:
Component 1: 1.7263
```

Figure 5.7 Average Path Length Result

The original graph's disconnection implies the existence of isolated subgroups within the network, preventing the computation of a singular average path length. However, within the anonymized graph, the average path length for connected components is calculated to be 1.7263. This relatively short average path length within connected components indicates that information can traverse between users efficiently within these subgroups. The presence of multiple connected components underscores the modularity of the network, as evidenced by the community detection results. Enhancing interconnections between these components could further reduce the overall average path length, promoting more efficient information flow across the entire network.

G. Top Connected Users

Identifying users with the highest number of connections can indicate influential members within the network. These users often play significant roles in information dissemination and community engagement.

Top 10 Users by Degree:

	username	degree
138	d*******_	229
165	s**********6	211
148	a*******a	198
3	s*************o	175
25	a******p	173
82	s*******0	167
38	m**************o	165
96	h************n	165
39	f********r	162
147	s******s	158

Figure 5.9 Top 10 Users by Degree

The top ten users exhibit significantly higher degrees of connectivity compared to the average user, with Node 139 (censored as d*******_) leading with 229 connections. These users are likely to be central figures within the student community, actively engaging with a large number of peers. Their extensive connections position them as key influencers who can effectively disseminate information, promote events, and foster community engagement.

VI. CONCLUSION

This study effectively applied graph theory to analyze the social network of the Himpunan Mahasiswa Informatika (HMIF) Institut Teknologi Bandung's Class of 2023 on Instagram. By utilizing the NetworkX library, the research meticulously constructed and examined the network's structural properties through centrality measures, community detection, network density, and average path length.

The analysis identified key influencers within the network, notably Nodes 139, 166, and 149, who possess high degrees of connectivity and play pivotal roles in information dissemination. Community detection revealed ten distinct subgroups, highlighting clusters of students with shared interests or close-knit relationships. The network density of 0.2631 indicated a moderately interconnected network, while the average path length within connected components demonstrated efficient information flow.

These findings offer valuable insights for enhancing student engagement and fostering a cohesive community. Leveraging influential nodes can optimize information dissemination and event participation, while understanding community structures allows for targeted engagement strategies tailored to specific student groups. Additionally, improving network connectivity can bridge isolated subgroups, promoting a more unified and resilient student network.

In summary, the integration of graph theory with NetworkX provided a robust framework for dissecting and understanding the social dynamics of the HMIF Class of 2023 on Instagram. The insights garnered from this analysis lay the groundwork for strategic initiatives aimed at enhancing student interactions and building a more interconnected academic community.

VII. APPENDIX

The source code for the implementation discussed in this paper is available at the following GitHub repository: https://github.com/l0stplains/SocialNetworkAnalysisHMIF23

VIII. ACKNOWLEDGMENT

The author wishes to express heartfelt gratitude to God for His guidance and support throughout the completion of this project. Sincere thanks are extended to the lecturers of ITB's Discrete Mathematics IF1220 course, Mr. Rinaldi Munir and Mr. Rila Mandala, for their invaluable knowledge and mentorship. Additionally, the author is deeply thankful to family and friends for their unwavering encouragement and support during this endeavor and chaotic end of semester. Their continuous support served as a foundation for overcoming challenges and achieving the objectives of this study.

REFERENCES

[1] Harary, F. *Graph Theory*. Addison-Wesley, 1969.

[2] Wasserman, S., & Faust, K. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

[3] Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 10, 2008. [Online]. Available: https://www.researchgate.net/publication/1913681_Fast_Unfolding_of_Communities_in_Large_Networks. Accessed: Jan. 5, 2025.

[4] Newman, M. E. J. (2004). Analysis of weighted networks. *Physical Review E*, 70(5), 2004. [Online]. Available: https://www.researchgate.net/publication/8128661_Analysis_of_Weighted_Networks. Accessed: Jan. 5, 2025.

DECLARATION OF ORIGINALITY

I hereby declare that the paper I have written is entirely my own work, and not a reproduction, adaptation, or translation of another individual's work. Furthermore, I declare that this paper is free from any form of plagiarism.

Bandung, 8 January 2025

Refki Alfarizi
13523002