SOCIAL NETWORK ANALYSIS OF HUAWEI COMMUNITY PLATFORM: IDENTIFYING KEY MEMBERS AND COMMUNITY STRUCTURES THROUGH CENTRALITY MEASURES, COMMUNITY DETECTION, AND CLUSTERING ALGORITHMS

[1]Njo, Rommy Haryanto, [2]Shinta Estri Wahyuningrum
[1,2] Program Studi Teknik Informatika Fakultas Ilmu Komputer,
Universitas Katolik Soegijapranata
[2]shinta@unika.ac.id

ABSTRACT

This research contains the application of the SNA (Social Network Analysis) algorithm to find out the most influential people or communities on the social media platforms that are Facebook, Twitter and Instagram Huawei pages. Social network data is taken from the Kaggle site entitled "Huawei Social Network Data", this data will be used for social media analysis. Currently, many of the most effective ways to do marketing are through 3 or more social media platforms, especially as large companies like Huawei definitely do this because their company coverage reaches all over the world. However, marketing online cannot be done haphazardly, because it will waste time and cost a lot of money. One of the factors that determines whether marketing is effective or not is the target market. Considering the importance of this, this research aims to find a good target market. One way to find a good target market is to use SNA. This research will use SNA algorithms such as , Centrality Measures, Community Detection, Clustering and other SNA algorithms. In the data there are 1000 columns and 1000 rows which are nodes and edges. The nodes labeled here are people's names and the number 1 is the number of edges. By analyzing this research, we can find out people or communities who have high potential to buy or even subscribe to Huawei products.

Keywords: Network, Sosial Network Analisys (SNA), Centrality Measures, Community Detection and Clustering

1 INTRODUCTION

1.1 Background

In today's digital era, marketing is not only done offline but also online. Online marketing is very effective if you use social media platforms such as Twitter, Facebook and Instagram. On social media platforms there is a lot of interaction between users so that information is easily spread. A lot of information is spread quickly and widely across the world, which is what makes companies want to do marketing on social media so that product information can also be spread. According to data from ILS 2016, The share of the world's population with Internet connection reached 46.1% in 2016 and according to data from Statista 2022a This share increased to 59.5% at the beginning of 2021 [11]. So therefore, digital marketing cannot be done haphazardly, if it is done the results will be less than optimal and will waste a lot of time and money for large companies like Huawei. Therefore, we need a systematic strategy for carrying out digital marketing.

In order to compete, companies must have an effective marketing strategy on social media platforms. One strategy is to find a good target market, so that the costs incurred are not in vain. The first thing you have to do is find out what social networks exist on a social media platform, for example Facebook. By knowing this network, we can analyze it using the SNA algorithm to find people or communities that have a lot of influence so that they can influence the people or communities connected to them. That is what constitutes a good target market and will make our marketing effective. SNA is a method for analyzing individual relationships in a network [2]. The SNA algorithm will calculate the node that is most connected to other nodes, that is the person who has the most influence in the community.

In this research, the dataset in the form of Matirx will be converted into a network for analysis. The algorithms used are Degree Centrality, Eigenvector Centrality, Betweenness Centrality and community detection on 3 platforms, namely Twitter, Facebook and Instagram, after which they will be analyzed and compared. Apart from that, the author also visualizes it so that it makes it easier for readers to understand. With the results of this data, we can find a good target market.

1.2 Problem Formulation

1. In this case, which community is best for marketing the product and who is the most influential person in this case?
2. Provides insight into optimizing Huawei's social media marketing strategy by effectively targeting the right audience?
3. How does SNA help find a good target market?
4. What are the most influential factors in marketing efficiency?

1.3 Scope

This research only finds people or communities that are good for marketing products. This research only calculates the data in the dataset, namely "Huawei Social Network Data" which contains Twitter, Facebook and Instagram data. Other things such as gender, salary, and what people in the target market are like are not analyzed here.

1.4 Objective

The aim of this research is to find people or communities with the highest potential for the target market in this case. The use of the SNA algorithm described in this research will provide an overview of the scientific perspective of SNA in solving this case.

2 LITERATURE STUDY

Amalia Ristantya et al. [1] has researched positive and negative perceptions based on comments on Instagram social media, they use the concept of social media analysis in their research. They use text network analysis and community detection. They divided their dataset into two positive comments and negative comments, the results they found that the social network was superior. Even though we apply the SNA concept, there are still many SNA algorithms that are not used. They also did not calculate centrality measures so the research was less accurate. The dataset is also only from the social media platform Instagram.

SNA is also used to analyze user interactions on social media regarding industry Fintech by Alisya Putri Rabbani et al. [2]. In this research they used SNA to analyze the SCRM (Social Customer Relationship Management) network in the fintech industry, namely OVO, Gopay and Linkaja. This research uses Density, Modularity, Average path length, Average degree, Reachability algorithms and other SNA algorithms. However, even though they are entitled about SNA, they do not identify the metrics that cause the SNA algorithm, such as Centrality Measures, Community Detection and clustering, so that the results are less accurate than previous research. The good thing about this research compared to previous research is that the dataset is sufficient to carry out analysis.

Research using SNA was also carried out for the dissemination of culinary information on social media by Yunila Dwi Putri Ariyanti [3]. The SNA algorithm used is more complete than the first and second studies, namely degree centrality, closeness centrality, betweenness centrality, and eginvector centrality. But the problem is that research only goes as far as calculating that, and not doing clustering. Apart from that, the dataset is only from the Twitter media platform, even though this research is quite new, namely 2022.

Apart from that, there is also another study that uses SNA and also analyzes the social media platform Twitter.This research contains Smartfren Social Network Analysis on Twitter Social Media, the method used is Text Network Analysis (TNA) which is actually the same as SNA only the way to define the network is based on text written by Delia Hayyuning Tyas et al. [4]. The algorithm used is Degree, Closeness, and Betweenness, The problem is still the same as before, namely the dataset is not good and the use of SNA is not enough. For comparison, this research was conducted in 2020, 2 years earlier than the research above.

Before the above research was carried out, there was research using SNA which was certainly interesting. The research was written by Risky Aswi R et al [5]. They analyzed the use of social networks to determine student concentration using the SNA method. What makes this research interesting is that the author added value variables to the analysis and combined them with SNA to increase the accuracy of the results in the form of student decisions. However, there are still many shortcomings in this research, namely the incomplete use of SNA and the lack of datasets, the same as previous studies.

Muchamad Taufiq Anwar et al. [6] also used SNA in his research to determine the distribution patterns of pornography on social media. There is a new algorithm in this research, namely Outdegree, which is not calculated by the studies mentioned previously. One of the advantages of this research is making comparisons and finding certain patterns in the pornography distribution network. But even so, the algorithm used is incomplete, namely only Degree Centrality, Betweenness Centrality and Reciprocity.

During the 2020 Depok city election campaign, there was monitoring of the use of Twitter social media using SNA. This research was written by Wildhan Khalyubi et al. [7]. What makes this research need to be used as literature other than an interesting case study is the use of SNA as monitoring as well as the use of clustering to see in which areas regional head candidates are superior or lacking. But unfortunately, just like the research mentioned above, the drawback is that the dataset is only from Twitter and the use of SNA is very minimal.

From the research above, it can be concluded that the use of SNA can be used in various types of cases. And this research is also different from the previous one, research conducted by Anggi I. Purba et al. [8]. SNA was used for social impact mapping in this research. In research, the author managed to find that price changes were the most important impact on the impact of Covid-19. However, the flaws are almost the same as previous studies, the SNA algorithm used only a small amount and did not perform clustering. even though this research is still relatively new, namely in 2023.

There is another research on SNA written by Made Kevin Bratawisnu and Andry Alamsyah [9]. His research contains analysis of user interactions regarding e-commerce businesses which have developed rapidly in recent years. What can be learned from this research is that they both used almost complete SNA and used sufficient datasets, namely from three e-commerce sites (Lazada, Tokopedia and Elevania). Apart from that, they also did not carry out community and clustering, even though if the research continued it would be better.

This research literature will then take research that shows how SNA can help marketing. This research was written by Ahmad Rifa'i [10]. In this research, use can provide results on which actors are most active and that will be the determining factor in decision making in carrying out promotions or target markets. The visualization in this research is good and can be used as an example. Research is almost the same as the research mentioned above, this research is still lacking, even though in this research they carried out clustering, the SNA algorithm used is still lacking. Another dataset is only from social media Twitter.

Furthermore, there is research written by Jeremiás Máté Balogh and Tamás Mizik [11] showing the importance of digital marketing in today's digital era. In this study they both did not use SNA, but they collected data from the top 12 wine shops in Hungary. In this study they both did not use SNA, but they collected data from the top 12 wine shops in Hungary and compared them. Because they don't use SNA and clustering algorithms, they don't find out what content has the biggest impact on wine shop growth.

Looking at the research above, research will be carried out on SNA in full, starting from identifying nodes and edges, then carrying out Centrality Measures followed by community detection and clustering for each network and finally comparing the three social media networks from the calculations carried out. Not only that, in this research the author used an adequate social media dataset to carry out the analysis. After the analysis is complete, one social media network will be compared with other social media networks. After doing that you will see which network has the most potential to reach a large market.

3 Research Methodology

3.1 Dataset Collection

The data in this research was taken from the Kaggle site entitled "Huawei Social Network Data", this dataset contains names of people and relationships between people taken from social media platforms that are Facebook, Twitter and Instagram Huawei pages. This data contains 3, namely data Facebook, Instagram and Twitter data, each of which contains 1001 columns and rows shaped like a matrix.

https://www.kaggle.com/datasets/andrewlucci/huawei-social-network-data/data

3.2 Data Pre-processing

The data in the form of a binary table is then taken as part of the data, namely 100 columns and rows for the next stage. The data is divided by one hundred to make observation and understanding easier when visualized. Except Centrality Measures which do not require data splitting because the output only contains numbers. The entire process can be seen in Figure 3. 1 The 10 data attachments in Table 3. 1 can be used to provide readers with a general overview of the data structure.

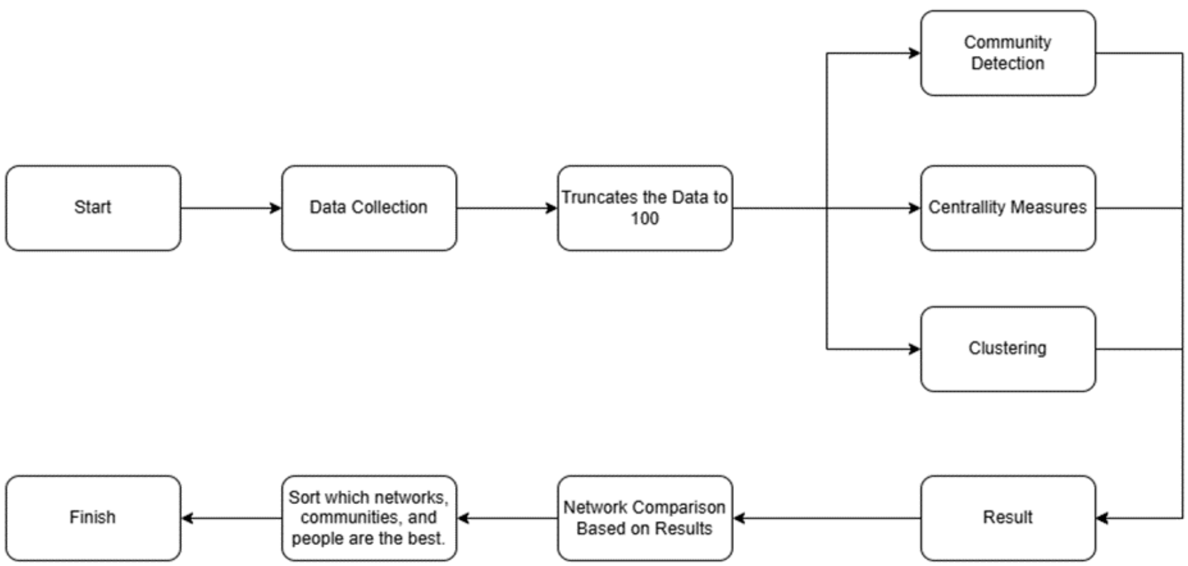

Figure 3. 1 Research Flowchart

Table 3. 1 Dataset Attachment

	MS	BM	YC	PD	SS	LC	ML	LL	KS	SA
MS	0	0	1	1	0	0	1	0	1	1
BM	0	0	1	0	0	0	1	1	0	0
YC	1	1	0	0	1	1	0	0	1	0
PD	1	0	0	0	0	1	0	1	1	0
SS	0	0	1	0	0	1	0	1	1	1
LC	0	0	1	1	1	0	1	1	1	0
ML	1	1	0	0	0	1	0	1	1	1
LL	0	1	0	1	1	1	1	0	1	0
KS	1	0	1	1	1	1	1	1	0	0
SA	1	0	0	0	1	0	1	0	0	0

This table contains names of people and relationships expressed in binary form, 0 meaning there is no relationship and 1 meaning there is a relationship. The names in this table are shortened because they are too long and the full names are in the table below.

Table 3. 2 Table Description

Abbreviation	Full Name
MS	Meredih Stransky
BM	Brittney Mazzella
YC	Yi Cook
PD	Porter Devries
SS	Suzanne Syverson
LC	Ladawn Creason
ML	Mikel Lamberson

Table 3. 2 serves to provide the user's full name which has been abbreviated in **Table 3. 1**. Hopefully this report can give readers an idea of the dataset format used. The type of graph in this dataset is an undirected and unweighted graph because the information is only 0 and 1.

So the way to create a graph from this dataset is to look at the column containing the username. This username will represent the node in the graph that will be formed later. Furthermore, the numbers 0 and 1 will indicate whether the node is connected or not 1 means connected and 0 means not connected. The table in this dataset is in the form of a matrix so that column 1 and row 1 (counted after the column) must be 0 because the node cannot be connected to itself. The writing (0,0) below means that the edge comes from column 0 and row 0. Therefore, from the example of the 10 x 10 table above, if the graph is changed, the results can be see in **Figure 3. 2** :

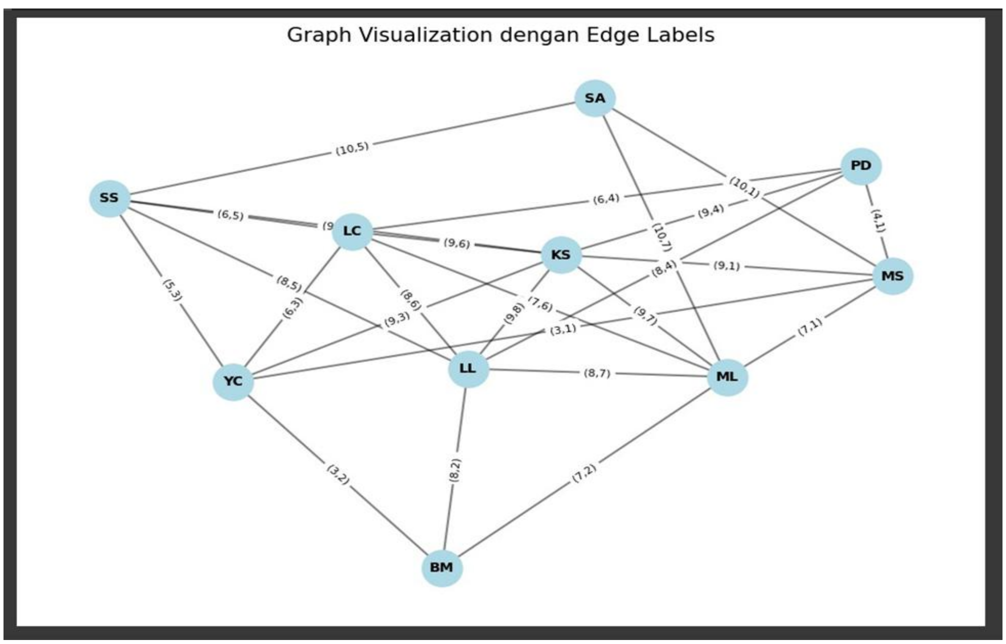

Figure 3. 2 Convert Table

To set up this research, the first thing to do is download the three datasets in .xlsx format and save them on Google Drive so they can be uploaded to Google Collab. "Furthermore, when the dataset has been downloaded, the next step is to install the Pandas library and use it to upload the downloaded dataset. Below is the coding to upload it.

```
import pandas as pd
twitter=pd.read_excel('/content/drive/MyDrive/SNA/Twitter_Data.xlsx')
instagram = pd.read_excel('/content/drive/MyDrive/SNA/Instagram_Data.xlsx')
facebook=pd.read_excel('/content/drive/MyDrive/SNA/Facebook_Data.xlsx')
```

Next, after all the data has been uploaded, we need to install all the libraries needed to run the SNA algorithm and those needed for visualization. The libraries used are pycairo, cairocffi, igraph, numpy, matplotlib, scipy and seaborn. To install it, you can use the coding below here.

```
!pip install pycairo
!pip install cairocffi
!pip install igraph
import igraph as ig
import cairocffi as cairo
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.sparse import linalg
import scipy.stats as stats
import seaborn as sns
```

After all the required libraries have been installed, we can proceed to the final stage, namely, changing the dataset format and cutting the dataset to 100 so that it can be calculated using the SNA algorithm and can be visualized properly. To change the format, the author uses Numpy by converting the Boolean data in the dataset into an Input List and Adjacency Matrix. However, specifically for the Centrality Measures algorithm, there is no need to split the data, therefore here the variables are separated. An example of the coding is below.

```
nama_jaringan_numpy = nama_jaringan.iloc[:, 1:].to_numpy()
nama_jaringan_graph = ig.Graph.Adjacency(nama_jaringan_numpy.astype(bool).tolist())
nama_jaringan2 = nama_jaringan[nama_jaringan.columns[1:101]]
nama_jaringan3 = nama_jaringan2.head(100)
nama_jaringan4 = nama_jaringan3.to_numpy()
```

3.3 Centrallity Measures, Clustering and Community Detection

After the data preprocessing has been completed, we move on to the next stage which is the core of this research. There are 5 Centrality Measures used in this research, namely Global Clustering, Local Efficiency, Degree Centrallity, Betweeness Centrallity and Closeness Centrallity. Then for Community Detection we use the Label algorithm Propagation. The last one is clustering, which is a more complicated algorithm than Community Detection using the walktrap algorithm.

1. Centrallity Measures

Centrality Measure is a calculation used to measure how much influence a node has on other nodes based on the edges connected to that node. Centrallity Measures has many algorithms, namely Degree Centrallity, Closeness Centrallity, Betweeness Centrallity and many more. For undirected graphs in this research, there are only 5 suitable algorithms, namely Global Clustering Coefficient, Local Efficiency, Degree Centrallity, Closeness Centrallity and Betweeness Centrallity. These algorithms will be explained below:

a. Clustering Coeficient

This Global Clustering Coefficient works by calculating the structure and number of edges in a network globally. A high Global Cluating Coefficient value indicates that many nodes are well connected to their neighbors and many triangles are formed. The formula is below, namely in **Equation 1**.

$$C = \frac{3.Triangle\ Count}{Triplet\ Count}$$

Equation 1 Clustering Coeficient

Description :

- Triangle: 3 connected nodes form a triangle,
- Triplets : Three connected nodes forming a triangle and a two-sided triangle.

b. Degree Centrallity

Degree Centrallity is the simplest algorithm in social network analysis. This algorithm works by counting the number of edges connected to a node. Based on this explanation, it can be concluded that if a node has a large Degree Centrality value then that node has a large influence too. The formula is below, namely in **Equation 2**.

$$C_{D(v)} = \frac{\deg(v)}{N - 1}$$

Equation 2 Degree Centrallity

Description :

- CD(v) : Degree Centrality of node v,
- deg(v) : Number of edges connected to node v,
- N : The total number of nodes in the network,
- N−1 : N minus node *v*.

c. Closeness Centrallity

Just like the name "Closeness", this algorithm calculates the closeness of a node to other nodes. This algorithm works by calculating the average distance from one node to another node. The smaller the distance between a node and other nodes, the easier it will be for information to spread. Then, the formula is below namely **Equation 3.**

$$C_{C(v)} = \frac{n-1}{\sum_{u\neq v} d(v,u)}$$

Equation 3 Closeness Centrallity

Description :

- n : Total number of nodes,
- d(v,u) : The shortest distance between nodes v and u,
- ∑_(u ≠v) d(v,u) : The number of steps required for node *v* to travel to all other nodes.

d. Betweeness Centrallity

If information spreads in a social network, there must be nodes that act as intermediaries for other nodes. This can be seen using the shortest path calculation from node to other node. Nodes that have an intermediary role are considered important by this algorithm, which is the Betweenness Centrality calculation. Then what is in **Equation 4** is the formula.

$$C_{B(v)} = \sum_{\{s\neq v\neq t\}} \frac{\sigma_{\{st\}(v)}}{\sigma_{\{st\}}}$$

Equation 4 Betweeness Centrallity

Description :

- v : Calculated nodes,
- s : Starting node,
- t : Destination node,
- σst : The number of shortest paths between nodes s and t,
- σst(v) : The number of shortest paths between nodes *s* and *t* through *v*.

e. Local Efficiency

Local Efficiency is an algorithm that measures how well a structure is formed locally. This algorithm is the opposite of the Global Clustering Coefficient which calculates the whole. Then what is in **Equation 5** is the formula.

$$E_i = \frac{1}{|N_i|(|N_i| - 1)} \sum_{j,k \in N_i, j \neq k} \frac{1}{d_{jk}}$$

Equation 5 Local Efficiency

Description :

- Ni : Neighbor of node i,
- djk : shortest distance from node *j* to *k*.

2. Clustering

Clustering is the process of grouping an entity, in this case social media users or nodes, based on existing similarities. The similarity in this case study is an edge, if a node has an edge connected to a node and its neighboring nodes also have edges connected to that node too, then that node and its neighbors can be identified as a cluster. The Walktrap algorithm is able to carry out clustering in a network that solid edge. The Walktrap algorithm formula has 3 stages, the first is Transition Matrix, Random Walk Probability Distribution and Community Similarity which can be seen in the figure below. **Equation 6** contains the Transition Matrix formula, then **Equation 7** contains the Random Walk Probability Distribution and finally **Equation 8** contains the Community Similarity formula.

1. Transition Matrix

$$P(i,j) = \frac{A(i,j)}{deg(i)}$$

Equation 6 Transition Matrix

Desciption :

- *P*(i,j) : Probability of moving from node i to node j in one step,
- A(i,j) : *Adjacency matrix of the graph, equal to 1 if there is an edge between nodes i and j, 0 otherwise.*
- *deg(i) : Degree of* node i, which is the number of immediate neighbors of i (sum of row *i* in *A*).

2. Random Walk Probability Distribution

$$P^t(i,j)$$

Equation 7 Random Walk

Description :

- t : *total* random walk specified.

3. Community Similarity

$$d(C1, C2) = || PC1 - PC2||^2$$

Equation 8 Community Similarity

Description :

- C : communities that exist in the network,
- d : dist*ance* between two community *C1* and *C2*.

3. Community Detection

To detect communities on this network, the author uses the Label Propagation algorithm. This algorithm can run on Google Collab, the IGraph library and is also compatible with the format of the dataset taken. This algorithm does not require initial assumptions/parameters because this algorithm will form a natural community based on the local network structure. then what is in **Equation 9** is the formula.

$$L_u = arg\ max_l \sum_{u \in N(u)} \delta(Lu, l)$$

Equation 9 Community Detection

Description :

- $N(u)$: The neighbors of node u,
- δ(Lv,l)$\delta(Lv,l)$: The delta function, which takes the value 1 if the label Lv=l and 0 otherwise.

Bab 3.4 Comparing the Networks of the three Social Media Platforms

After getting the data from the calculations, we will compare the three platforms, namely Twitter, Facebook and Instagram. That way we can know which network is the best for doing digital marketing. Which is explained in more detail in **Figure 3. 11**.

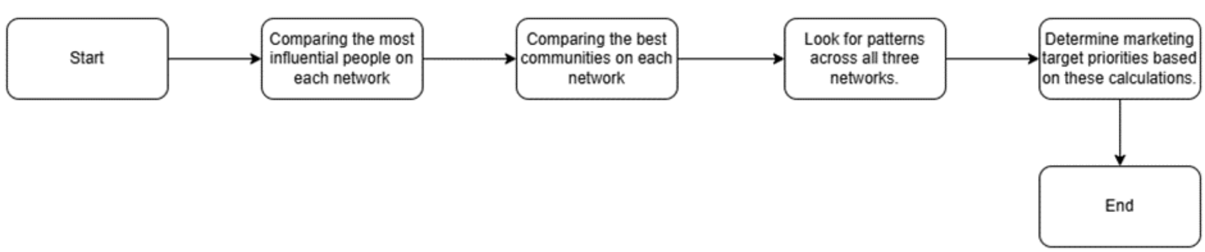

Figure 3. 3 Comparing Networks

4 IMPLEMENTATION AND RESULT

4.1 Data Colecting

This data was taken from Kaggle entitled " Huawei Social Network Data " then downloaded in xls form and imported into Google Collab using the Pandas library. Don't forget to save the three dataset files on Google Drive which is integrated with Google Collab and give the folder a name that is easy to remember.

4.2 Experiment Setup

In this research, the author used Google Colab, more specifically Python and the Igraph library to calculate it. First, the author imported the Pandas library to enter data that had been downloaded from Kaggle. Next, the author changed the dataset format to match the algorithm in the Igraph library. The coding that can be used is below or **Figure 4. 1**.

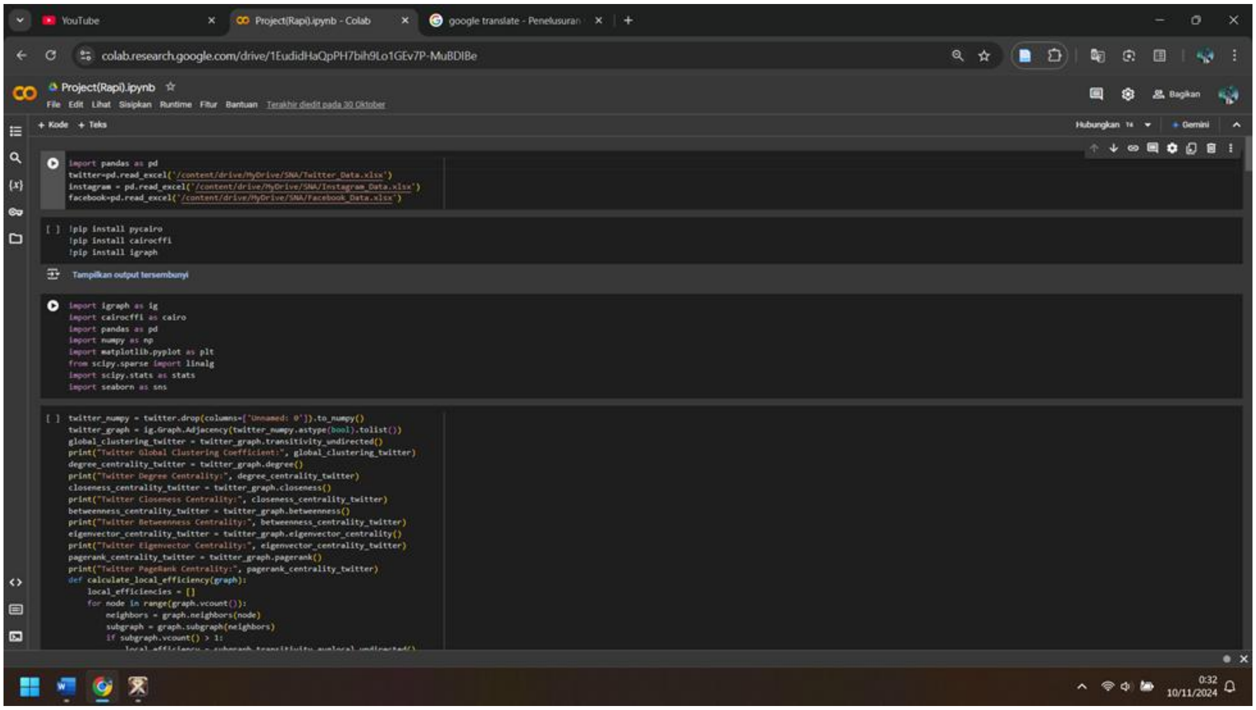

Figure 4. 1 Set Up

4.3 Centrallity Measures, Community Detection and Clustering

Entering the next stage, namely centrality measure and others, the author uses the library from igraph to calculate data with this algorithm. The following are details of the centrality measure used :

1. Global Clustering Coefficient, an algorithm that measures the extent to which nodes in a network tend to cluster together.

2. Degree Centrality, an algorithm that measures centrality based on the number of ties that exist between nodes.
3. Closeness Centrality, an algorithm that calculates the average shortest distance between each node.
4. Betweeness Centrality, an algorithm that measures how much influence a node has on the flow of information with other nodes.
5. Local Efficiency, an algorithm that measures the average length of the shortest path in an area in the network.

The following is the code for the centrality measures for all networks. Apart from calculating this code, it will also display the results in numerical form. The coding can be seen below which uses the igraph library and can be modified according to needs. These 5 centrality measures algorithms have been normalized so that the calculations are in accordance with the formula in CHAPTER 3.

```
def calculate_network_metrics(graph):
    n = len(graph.vs)
    max_possible_betweenness = (n - 1) * (n - 2) / 2
    betweenness_centrality = graph.betweenness()
    betweenness_centrality_normalized = [b / max_possible_betweenness for b in betweenness_centrality]
    avg_betweenness_centrality_normalized = sum(betweenness_centrality_normalized) / n
    closeness_centrality = graph.closeness()
    avg_closeness_centrality = sum(closeness_centrality) / n
    degree_centrality = graph.degree()
    degree_centrality_normalized = [d / (n - 1) for d in degree_centrality]
    avg_degree_centrality_normalized = sum(degree_centrality_normalized) / n
    clustering_coefficient = graph.transitivity_local_undirected()
    global_clustering_coefficient = graph.transitivity_undirected()
    def local_efficiency(graph):
        local_eff = []
        for v in range(len(graph.vs)):
            neighbors = graph.neighbors(v)
            if len(neighbors) < 2:
                local_eff.append(0)
            else:
                subgraph = graph.subgraph(neighbors)
                local_eff.append(subgraph.transitivity_undirected() or 0)
        return local_eff
    local_efficiency_values = local_efficiency(graph)
    avg_local_efficiency = sum(local_efficiency_values) / n
    return {
        'betweenness_centrality': betweenness_centrality_normalized,
        'avg_betweenness_centrality': avg_betweenness_centrality_normalized,
        'closeness_centrality': closeness_centrality,
        'avg_closeness_centrality': avg_closeness_centrality,
```

```
        'degree_centrality': degree_centrality_normalized,
        'avg_degree_centrality': avg_degree_centrality_normalized,
        'clustering_coefficient': clustering_coefficient,
        'global_clustering_coefficient': global_clustering_coefficient,
        'local_efficiency': local_efficiency_values,
        'avg_local_efficiency': avg_local_efficiency
    }
```

Below is the coding for clustering. The variables below can be replaced with the network name, for example Twitter, Instagram or Facebook. clusteringnga uses the walktrap algorithm which is suitable for this dataset. Then, if you want to see the visuals and modularity, you can use the coding below again.

```
clustering_namajaringan_walktrap = namajaringan_graph_100.community_walktrap()
clusters_namajaringan_walktrap = clustering_namajaringan_walktrap.as_clustering()
labels_namajaringan_walktrap = clusters_namajaringan_walktrap.membership
palette_namajaringan_walktrap = ig.RainbowPalette(len(set(labels_namajaringan_walktrap)))
num_communities_namajaringan_walktrap = len(set(labels_namajaringan_walktrap))
modularity_namajaringan_walktrap = clusters_namajaringan_walktrap.modularity
community_sizes_namajaringan_walktrap = [labels_namajaringan_walktrap.count(i) for i in set(labels_namajaringan_walktrap)]
average_members_namajaringan_walktrap = sum(community_sizes_namajaringan_walktrap) / len(community_sizes_namajaringan_walktrap)

  print(f"Jumlah Cluster Namajaringan (Walktrap): {num_communities_namajaringan_walktrap}")
  print(f"Modularitas Namajaringan (Walktrap): {modularity_namajaringan_walktrap}")
  print("Anggota Cluster per Grup Namajaringan (Walktrap):", community_sizes_namajaringan_walktrap)
  ig.plot(
      clusters_namajaringan_walktrap,
      vertex_label=namajaringan.iloc[:100, 0].tolist(),
      palette=palette_namajaringan_walktrap,
      vertex_size=20,
      layout=namajaringan_graph_100.layout("fr")
  )
```

The coding below is coding for community detection and the algorithm used is label propagation. The network name variable can be replaced by Twitter, Instagram and Facebook depending on which one you want to calculate. Don't forget, if you want to see the visuals, you can use the coding below again.

```
community_namajaringan = namajaringan_graph_100.community_label_propagation()
print(f"Jumlah Komunitas Namajaringan: {len(set(community_namajaringan.membership))}")
print("Anggota Komunitas per Grup Namajaringan:", [list(community_namajaringan.membership).count(i) for i in set(community_namajaringan.membership)])
vertex_labels_namajaringan = namajaringan.iloc[:100, 0].tolist()
ig.plot(
    community_namajaringan,
    vertex_label=vertex_labels_namajaringan,
    palette=ig.RainbowPalette(len(set(community_namajaringan.membership))),
    vertex_size=20,
        layout=namajaringan_graph_100.layout("fr")
   )
```

4.4 Result

So from the calculations that have been done, the result is that the Twitter network is the best to become a target market. Specifically, the Centrallity Measures results were made with the condition that they would be sufficient on this sheet of paper, namely by averaging all the results in each nodes 1000 of Centrallity Measures algorithms, except Global Clustering. Following are the detailed calculations.

1. Centrallity Measures

Table 4. 1 Centrallity Measures

Centrallity Measures	Network		
	Twitter	*Instagram*	*Facebook*
Clustering Coefficient (Global)	0.501201	0.008399	0.100290
Degree Centrallity (Average)	0,50113	0,00987587	0,100406
Closeness Centrallity (Avarage)	0.66725	0.306247	0.526431
Betweenness Centrallity (Avarage)	0.0004998686	0.00227769	0.000901
Local Efficiency (Average)	0.501237	NaN	0.100783

Table 4. 1 shows the results of calculating Centrality Measures from three networks, including Twitter, Instagram and Facebook. There are 5 algorithms used for Centrality Measures, namely Clustering Coefficient, Degree Centrality, Closeness Centrality, Betweenness Centrality, and Local Efficient. "The five algorithms have their respective functions.

The first algorithm is Clustering Coefficient. This algorithm's calculations will show how connected a node is to its neighbors. There are two types of Clustering Coefficient, namely Global Clustering Coefficient and Local Clustering Coefficient. In **Table 4. 1** we can see that the Twitter

Network has the largest Global Clustering value, worth 0.501201. This is because the nodes in the Twitter Network have many and far edges. then the lowest is Instagram Network, worth 0.008399 and this shows otherwise.

Next is Local Efficiency, which measures the connectivity of a node and its neighbors within a localized scope. Unlike Global Clustering, Local Efficiency is **calculated** individually for each node, resulting in unique values for each. Due to the excessive amount of output generated from the code, the author simplified the results by computing the average Local Efficiency across all nodes. The data presented in Table 4.1 represents this aggregated calculation.

The highest average Local Efficiency value is recorded in the Twitter Network, with a score of 0.501237. This indicates that each node is well-connected to its neighbors, forming numerous triplets. On the other hand, the Instagram Network has a NaN value because the number of edges is too small, preventing the algorithm from performing the calculation.

The third Centrallity Measures algorithm is Degree Centrallity. This algorithm is the easiest algorithm to understand. Simply put, this algorithm works by calculating how many edges a node has and then comparing it to the total nodes. then the results of the calculations show that the Twitter Network is the best network based on the average Degree Centrality value, which is 0.50113. Why average? because the output produced by the Degree Centrality calculation is the same as Local Efficiency, worh 1000. This output represents all the nodes in the network. For other networks the values are much lower, worth 0.00987587 and 0.100406. The data can be seen in **Table 4. 1**.

"Next is Closeness Centrality, the method of calculation is that nodes will be calculated based on **how** many edges are needed to travel to all other nodes. The coding produces 1000 outputs representing all nodes, so the Closeness Centrality data in **Table 4. 1** is normalized data. Based on **Table 4. 1**, the Twitter Network has the highest value, worth 0.66725, the same as before, followed by the second largest, namely the Facebook Network, worth 0.526431 and finally the Instagram Network, which has the lowest value of 0.306247.

The fifth algorithm is Betweeness Centrallity. Betweeness Centrallity calculates how often a node acts as an intermediary for information from other nodes. The data displayed in **Table 4. 1** is the result of calculating the average Betweenness Centrality value of all nodes of each network. The results are surprising where the Twitter Network which is usually the highest is now the lowest. The Twitter network is worth 0.0004998686, while the Instagram and Facebook networks are worth 0.00227769 and 0.000901. This happens because there are few nodes in the Twitter and Facebook networks that act as intermediaries for other nodes. Nodes in the network are connected directly to the destination node or j in the formula. Therefore, this does not mean that the Twitter and Facebook networks are bad. Even though the Instagram Network has the highest value compared to other networks, the value of 0.00227769 is low for Betweenness Centrality Value.

2. Clustering
a. Twitter Network

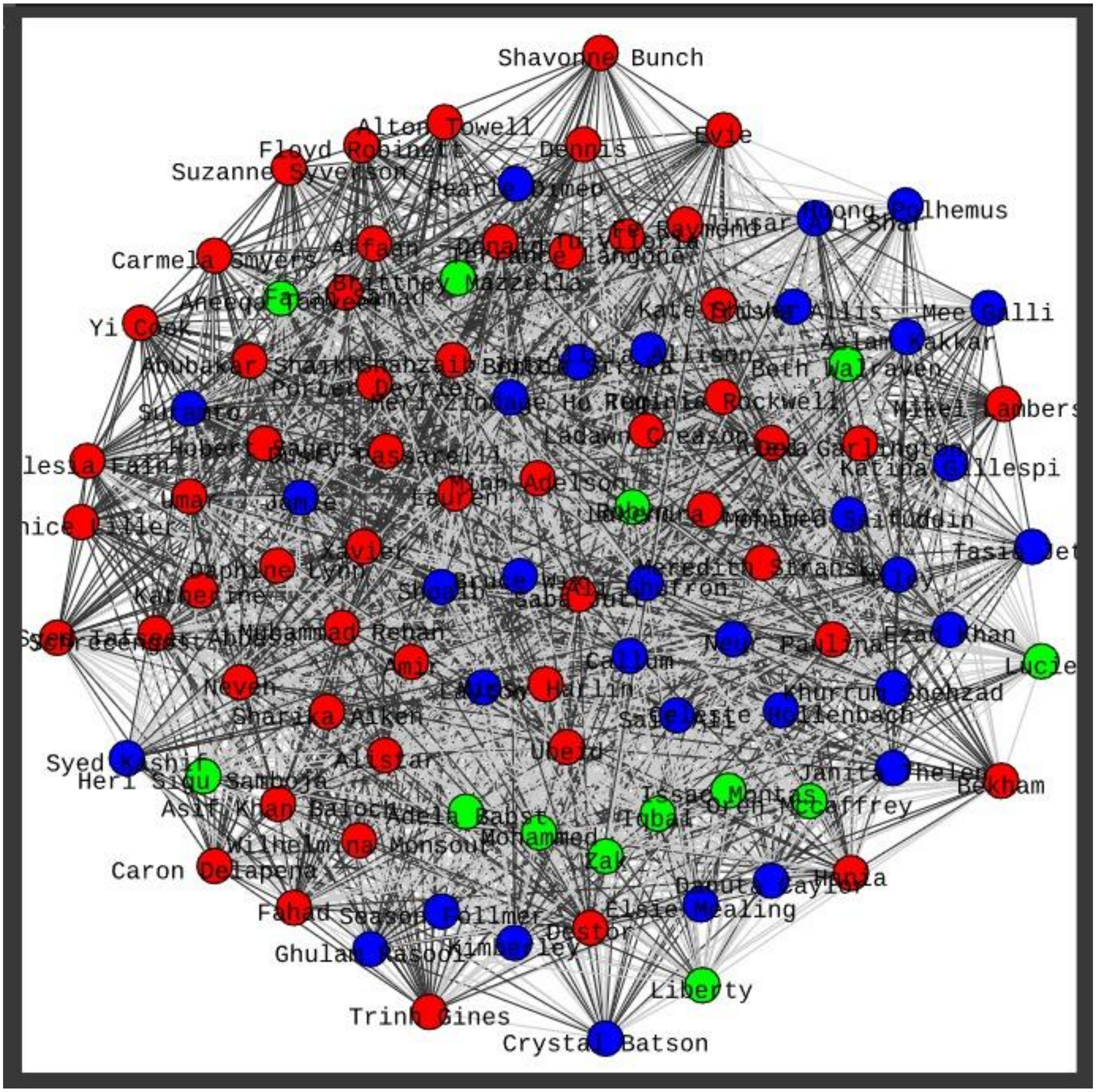

Figure 4. 2 Twitter Clustering

Figure 4. 2 is the result of Twitter Network clustering. Clusters are marked with different colors. The number of clusters formed from the Twitter Network is 3, then the average **number** of members is 33.333 with a modularity of 0.0545526226741192. Therefore, it can be said that the Twitter Network is the best compared to other networks based on clustering calculations.

b. Instagram Network

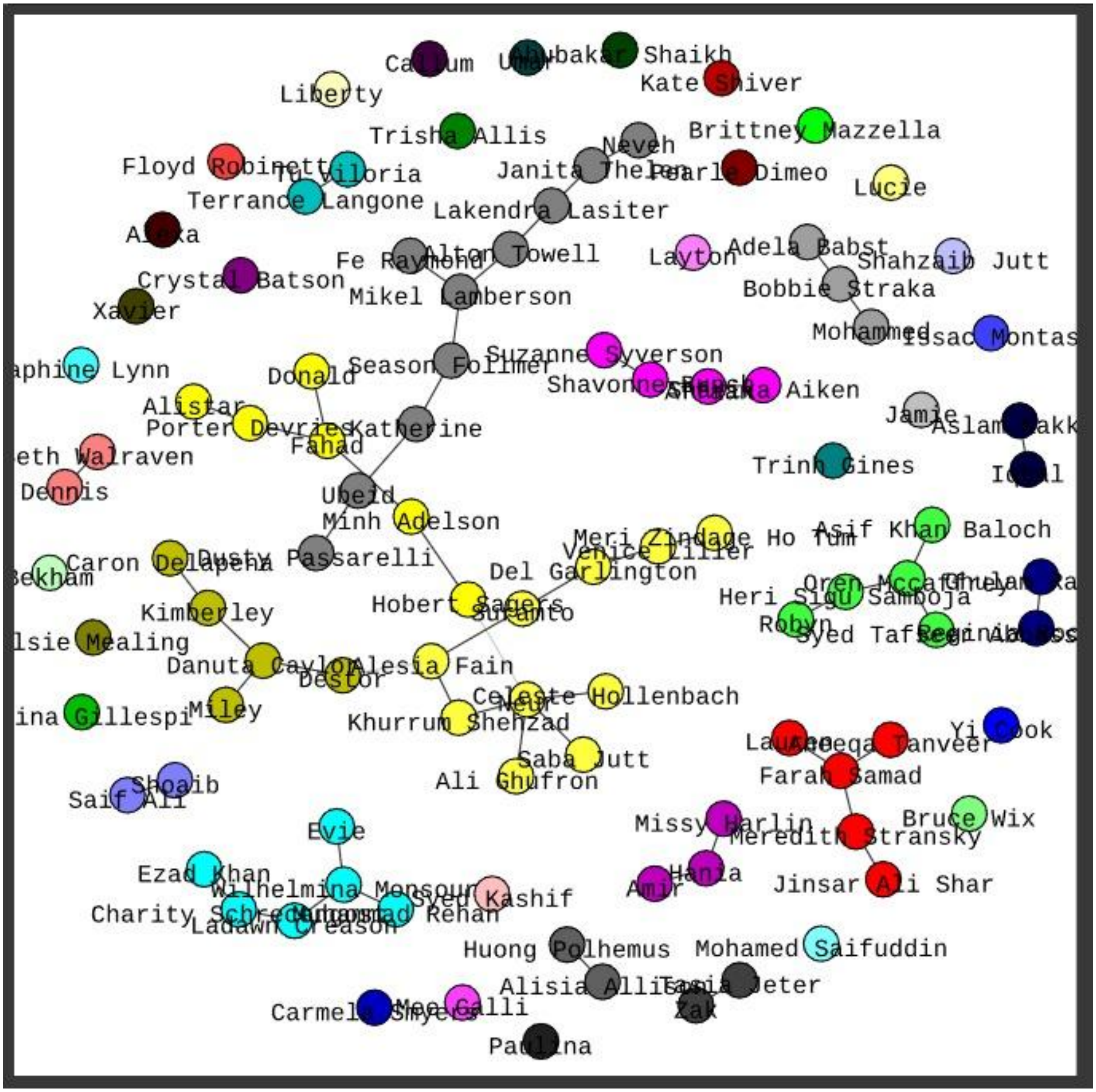

Figure 4. 3 Instagram Clustering

Then the Instagram Network, this Instagram Network has clear clusters which can be seen in **Figure 4.3**, so the modularity value is the highest compared to other networks, namely 0.**8646364795918365**. A high modularity value indicates that many nodes are only connected to one cluster. This causes the Instagram Network to be unsuitable as a target market because it will require a lot of resources to reach all users. The number of clusters is 46 and the average number of members is 2.17.

c. Facebook Network

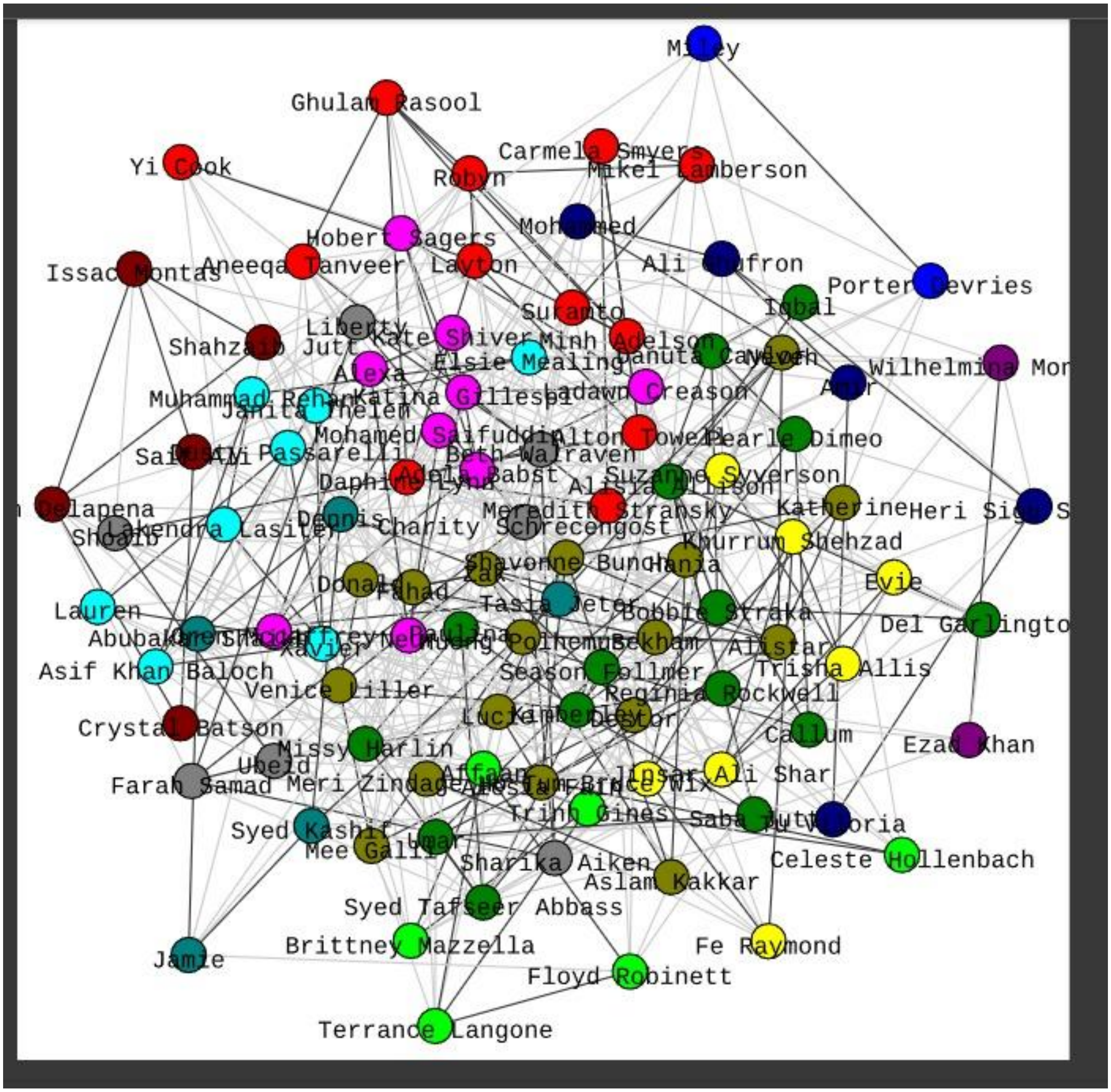

Figure 4. 4 Facebook Clustering

This Facebook network is in the middle, not as good as the Twitter network and also not as bad as the **Instagram** network based on the results of Clustering. As seen in the **Figure 4. 4**, the Facebook Network has as many edges as but not as many as the Twitter Network. Apart from that, the Facebook Network has an ideal number of clusters, not too many like Instagram, namely 13 clusters which are colored differently in the **Figure 4. 4**.

d. Comparison based on Clustering

Table 4. 2 Comparrison Based on Clustering

Network and Algorithm	**Clusters**	**Average members count**	**Modularity**
Twitter (Walktrap)	3	33.33	0.0545526226741192
Instagram (Walktrap)	46	2.17	0.8646364795918365
Facebook (Walktrap)	13	7.69	0.21716646547224916

In **Table 4. 2** you can see the clustering results using the walktrap algorithm. Clustering is the process of grouping nodes based on the edge structure formed. In contrast to community detection, Clustering does not focus on finding communities, but focuses on node and edge patterns. From the results that can be seen from **Table 4. 2**, the Twitter algorithm has the fewest clusters, worth 3 and an average number of members of 33.33. This is good because the data is not divided into several clusters. There is an important note here that the modularity value of the Twitter Network is the lowest, which is around 0.0545526226741192, meaning that many nodes are connected to more than 1 cluster so that the algorithm considers it outside the community, but that doesn't mean the Twitter Network is bad, likewise the Facebook Network is inferior to the Instagram Network. The Instagram network has the highest modularity, worth 0.8646364795918365 because on average the nodes in the network only have one cluster so the division of clusters is clear. The nodes in these three social media networks are set the same, namely 100 nodes, therefore the Instagram Network has the largest number of clusters and the lowest average number of members.

3. Community Detection

a. Twitter Network

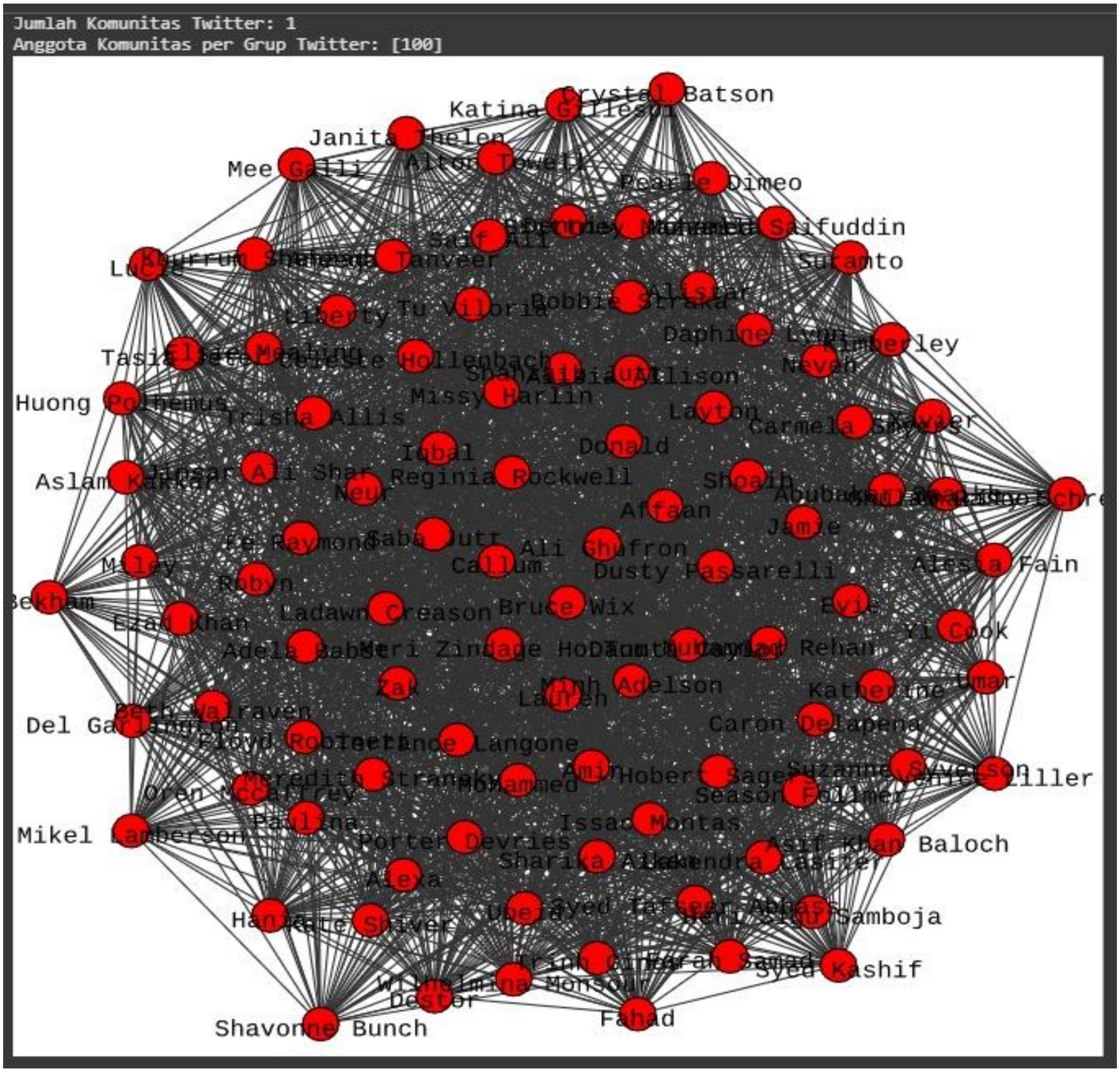

Figure 4. 5 Twitter Community

Figure 4. 5 above is a visualization image of the Twitter Network from the Community Detection algorithm, namely Label Propagation. This algorithm calculates by giving a label to each node, if the nodes are connected then the label is the same which is considered a community. It can be seen that the Twitter network has a very large number of edges, so that no community is formed, almost all nodes have many edges. Therefore, it can be concluded that the Twitter Network has 1 community with 100 members or all nodes.

b. Instagram Network

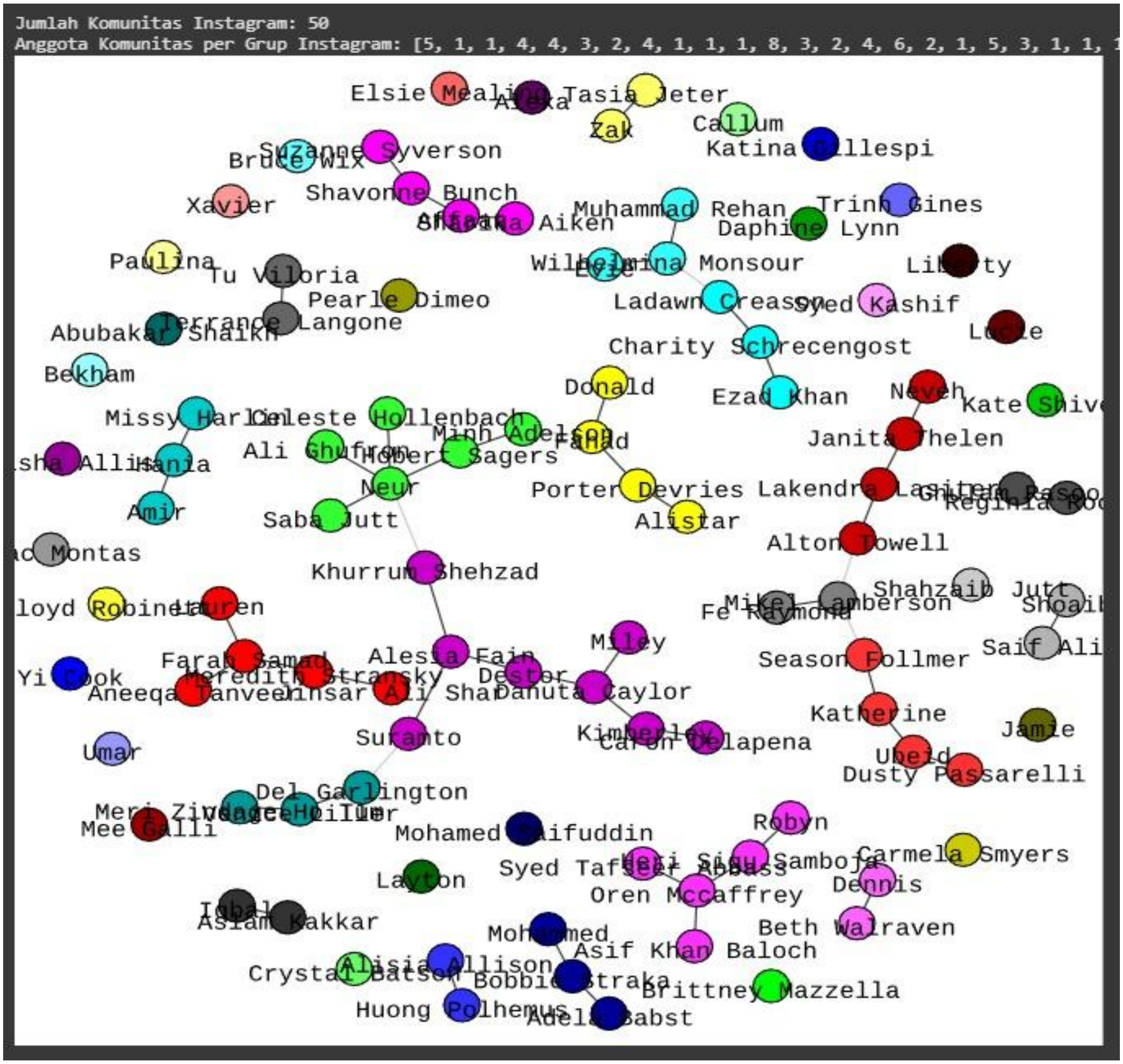

Figure 4. 6 Instagram Community

Next is the Instagram Network, the visualization results can be seen in **Figure 4. 6** and the number of communities formed is 50 and the average member of each community is 2. The edges in this network are not widely distributed to all nodes and tend to form communities. When compared to the Twitter Network, the Instagram Network is less centralized and the network is divided so that information does not spread widely.

There are some communities that only consist of one node, of course this cannot be called a community. Therefore the actual number of communities is 21 and the average number of members in each community is 3.38. Based on the results of these calculations, we can conclude that the Instagram Network is much worse than the Twitter Network, with a network structure like that, information between nodes will be difficult to spread, especially as there are nodes that are not included in the community.

c. Facebook Network

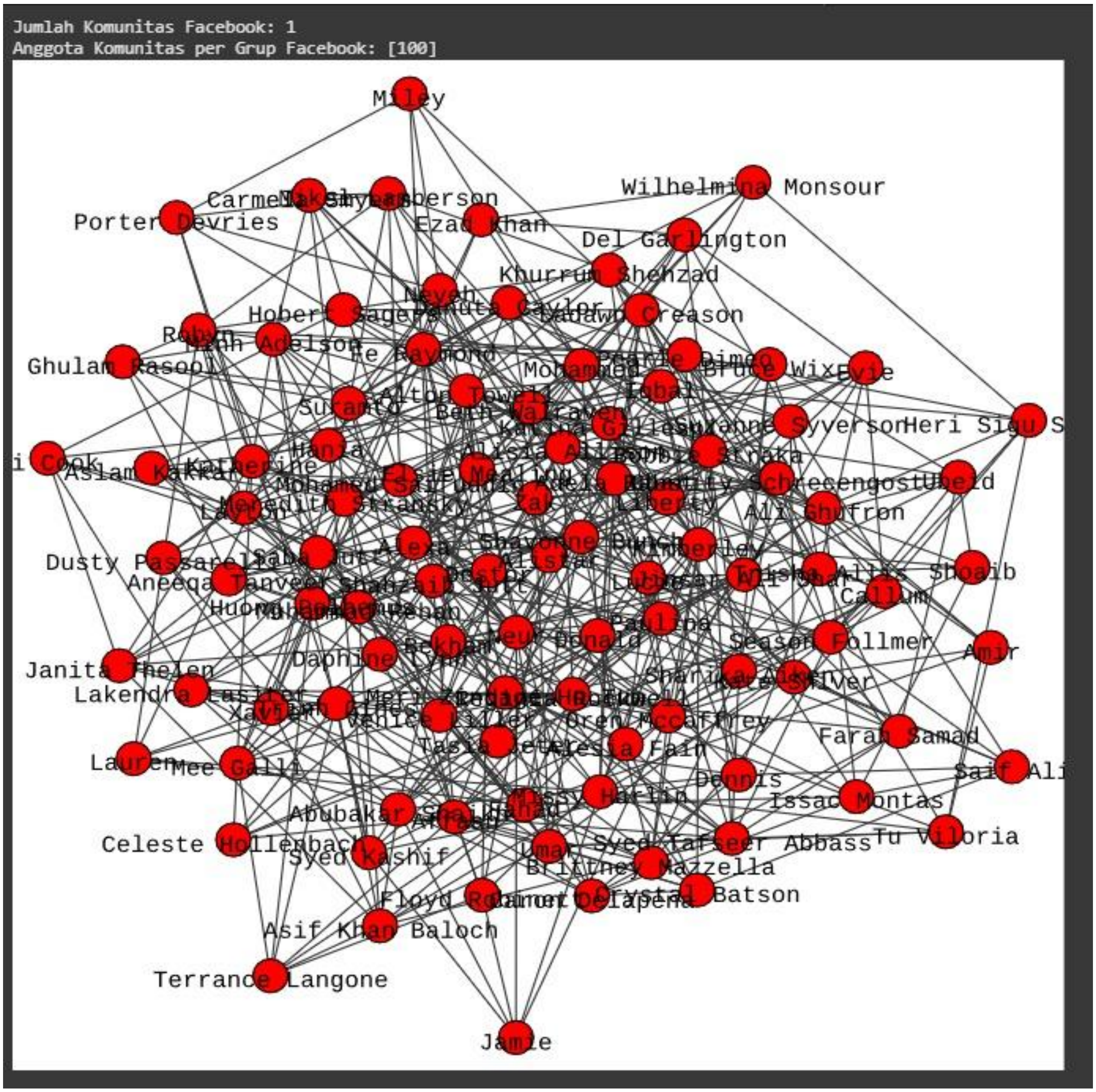

Figure 4. 7 Facebook Community

The last one is the Facebook Network. This Facebook network has 1 community and the average number of members is 100 based on the Label Propagation algorithm calculations seen in Figure 4. 7. This Facebook network can be said to be in the middle, not as good as the Twitter network and not as bad as the Instagram network. Although the number of communities is the same as the Twitter Network, the number of edges is less than the Twitter Network.

d. Comparison based on Community Detection

Table 4. 3 Comparison based on Community Detection

Network	Number of Groups	Total community members	Average members count
Twitter	1	100	100
Instagram	50	100	2
Facebook	1	100	100

Table 4.3 presents the comparison results of the three networks based on community detection calculations, using three parameters: the number of clusters, the number of members, and the average number of members. The Instagram Network has the largest number of communities. However, this is not necessarily a good thing, as information tends to remain confined within each community. Some nodes even exist in isolation without belonging to any community. If these isolated nodes are not considered communities, the number of communities is reduced to 21, with an average of 3.38 members per community.

Meanwhile, both the Twitter Network and the Facebook Network consist of a single community based on the propagation label algorithm calculation. However, unlike the Instagram Network, these two networks do not have isolated nodes. This is beneficial, as it ensures that the network remains connected, allowing information to spread more efficiently.network.

4.5 Discussion

The research results demonstrate Twitter Network's superiority across all algorithms employed, as evidenced by its consistently high values. This superiority can be attributed to the fundamental system of the social media platform. According to Muchamad Taufiq Anwar [6], Twitter's publisher page is accessible to all users, even those without accounts or following relationships. This open accessibility enables rapid information dissemination across the entire Twitter userbase, explaining why the Twitter Network achieves the highest values across all algorithm calculations, with the exception of Betweenness Centrality, as detailed in Chapter 4's Results section.

The Instagram network consistently shows the lowest average values compared to other networks, which can be explained by its Following and Follower system. This restrictive system limits information dissemination to only those users who are followers. Consequently, the Instagram Network proves unsuitable for marketing Huawei products, as it would require inefficient resource allocation for marketing to individual clusters, ultimately resulting in wasted time and efforts.

This research was limited to analyzing unweighted and undirected edges in the network due to the lack of diverse datasets. The available dataset only contained basic connection information between users, without additional attributes such as interaction frequency, communication type, or directional flow of information. With more varied data, the analysis could be significantly advanced. For instance, weighted edges could measure communication intensity between social media users, providing insights into relationship strength and influence patterns. If the graph were directed, it could reveal information flow patterns, identify opinion leaders, and determine which users act as primary information sources versus information consumers. These limitations stemmed from data collection constraints and the difficulty in obtaining comprehensive social media interaction data, particularly regarding user privacy and platform API restrictions. Future research would benefit from more detailed datasets that capture the complex nature of social media interactions.

5 CONCLUSION

The SNA algorithm is able to assess the structure formed from nodes and edges, not just the number of edges or just one node and a network can use well-conducted research for analysis and reference for further research. The most influential factor is the number of communities, the number of edges, then the next are the intermediate nodes in each community, that's why the SNA algorithm is needed. For the final result, the Twitter network is the best to be used as a target market based on value Centrality Measures, Clustering and Community Detection. So, market strategy suggestions based on the results of this research are first to focus on the social media Twitter because information will be spread throughout the network there, secondly to focus on nodes or users that have a high centrality value because the output is too much so it cannot be displayed here.

The graphs in this research are only undirected graphs, so for further research it is recommended to use other types of graphs. Examples of directed graphs, weight graphs or perhaps a combination of both. "For case studies, it is recommended to use case studies that are more important and newer than this research, for example the network in crypto, namely blockchain.

DAFTAR PUSTAKA

[1] A. Ristantya, V. Okiyani, E. Triyandah, F. Febrianti, S. Aninditha, and S. Kamalliah, "Analisa Interaksi Pelanggan dengan Konsep Social Network Analysis untuk Mengetahui Persepsi Positif dan Negatif berdasarkan Komentar pada Media Sosial Instagram," J-SAKTI (Jurnal Sains Komputer dan Informatika), vol. 4, no. 2, Art. no. 2, Sep. 2020, doi: 10.30645/j-sakti.v4i2.217

[2] A. P. Rabbani, A. Alamsyah, and S. Widiyanesti, “ANALISA INTERAKSI USER DI MEDIA SOSIAL MENGENAI INDUSTRI FINTECH MENGGUNAKAN SOCIAL NETWORK ANALYSIS (STUDI KASUS: GOPAY, OVO DAN LINKAJA),” Jurnal Mitra Manajemen, vol. 4, no. 3, Art. no. 3, Mar. 2020, doi: 10.52160/ejmm.v4i3.352

[3] Y. D. P. Ariyanti, “Analisis Centrality Aktor pada Penyebaran Informasi Kuliner di Media Sosial dengan menggunakan Social Network Analysis,” Journal of Systems, Information Technology, and Electronics Engineering, vol. 2, no. 1, Art. no. 1, Jun. 2022. Available: https://www.e-journal.ivet.ac.id/index.php/jsite/article/view/2306

[4] G. A. Suganda, H. H. Putri, M. Nurtikasary, and S. U. A.r, “Analisis Jejaring Sosial Smartfren Pada Sosial Media Twitter Dengan Metode Text Network Analysis (TNA),” J-SAKTI (Jurnal Sains Komputer dan Informatika), vol. 4, no. 2, Art. no. 2, Sep. 2020, doi: 10.30645/j-sakti.v4i2.218

[5] R. A. R, W. W. Winarno, and M. R. Arief, “ANALISIS PEMANFAATAN JEJARING SOSIAL UNTUK PENENTUAN KONSENTRASI MAHASISWA DENGAN METODE SOCIAL NETWORK ANALYSIS (Studi Kasus: Universitas Nusantara PGRI Kediri),” Telematika, vol. 8, no. 2, Art. no. 2, Aug. 2015, doi: 10.35671/telematika.v8i2.394. Available: https://ejournal.amikompurwokerto.ac.id/index.php/telematika/article/view/394. [Accessed: Apr. 16, 2024]

[6] M. T. Anwar, “Analisis Pola Persebaran Pornografi pada Media Sosial dengan Social Network Analysis,” Jurnal Buana Informatika, vol. 9, no. 1, Art. no. 1, Jul. 2018, doi: 10.24002/jbi.v9i1.1667

[7] W. Khalyubi, C. D. Bangun, M. R. Romadhona, and F. Ardiyansyah, “Pemantauan Penggunaan Media Sosial Twitter Menggunakan Social Network Analysis dalam Melihat Politik Programatik pada Kampanye Pilkada Kota Depok Tahun 2020,” Jurnal Adhyasta Pemilu, vol. 4, no. 1, Art. no. 1, Dec. 2021, doi: 10.55108/jap.v4i1.45

[8] A. I. Purba, I. Indra, and A. Nugroho, “Pemetaan Dampak Sosial dan Ekonomi Akibat Covid-19 Pada Stakeholder Kopi Aceh Tengah Menggunakan Pendekatan Social Network Analysis (SNA),” Jurnal Ilmiah Mahasiswa Pertanian, vol. 8, no. 2, Art. no. 2, May 2023, doi: 10.17969/jimfp.v8i2.24023

[9] M. K. Bratawisnu and A. Alamsyah, “SOCIAL NETWORK ANALYSIS UNTUK ANALISA INTERAKSI USER DI MEDIA SOSIAL MENGENAI BISNIS E-COMMERCE (STUDI KASUS: LAZADA, TOKOPEDIA DAN ELEVENIA),” Almana : Jurnal Manajemen dan Bisnis, vol. 2, no. 2, Art. no. 2, Aug. 2018, doi: 10.36555/almana.v2i2.143

[10] A. Rifai, E. P. Rachmawati, and D. A. Wicaksono, “Social Network Analysis dalam Mengukur Keaktifan Promosi Universitas di Media Sosial Twitter,” Indonesian Journal of Computer Science, vol. 11, no. 3, Art. no. 3, Dec. 2022, doi: 10.33022/ijcs.v11i3.3097. Available: http://ijcs.stmikindonesia.ac.id/ijcs/index.php/ijcs/article/view/3097. [Accessed: Jun. 29, 2024]

[11] J. M. Balogh and T. Mizik, “Impacts of Marketing Strategy and Social Media Activity on the Profitability of Online Wine Shops: The Case of Hungary,” Economies, vol. 10, no. 12, Art. no. 12, Dec. 2022, doi: 10.3390/economies10120301

APPENDIX

PREPROCESSING DATA

```
1. !pip install igraph
2. !pip install cairocffi
3. import pandas as pd
4. twitter=pd.read_excel('/content/drive/MyDrive/SNA/Twitter_Data.xlsx')
5. instagram = pd.read_excel('/content/drive/MyDrive/SNA/Instagram_Data.xlsx')
6. facebook=pd.read_excel('/content/drive/MyDrive/SNA/Facebook_Data.xlsx')
7. import igraph as ig
8. import cairocffi as cairo
9. import pandas as pd
10.   import numpy as np
11.   import matplotlib.pyplot as plt
12.   import networkx as nx
13.   from scipy.sparse import linalg
14.   import scipy.stats as stats
15.   import seaborn as sns
16.   import matplotlib.colors as mcolors
17.   twitter_numpy = twitter.iloc[:, 1:].to_numpy()
18.   undirected_matrix = np.maximum(twitter_numpy, twitter_numpy.T)
19.   twitter_graph = ig.Graph.Adjacency(undirected_matrix.astype(bool).tolist(), mode=ig.ADJ_UNDIRECTED)
20.   instagram_numpy = instagram.iloc[:, 1:].to_numpy()
21.   undirected_matrix_instagram = np.maximum(instagram_numpy, instagram_numpy.T)
22.   instagram_graph = ig.Graph.Adjacency(undirected_matrix_instagram.astype(bool).tolist(), mode=ig.ADJ_UNDIRECTED)
23.   facebook_numpy = facebook.iloc[:, 1:].to_numpy()
24.   undirected_matrix_facebook = np.maximum(facebook_numpy, facebook_numpy.T)
25.   facebook_graph = ig.Graph.Adjacency(undirected_matrix_facebook.astype(bool).tolist(), mode=ig.ADJ_UNDIRECTED)
26.   print("Jumlah node dan edge di Instagram:", instagram_graph.vcount(), instagram_graph.ecount())
27.   print("Jumlah node dan edge di Facebook:", facebook_graph.vcount(), facebook_graph.ecount())
28.   print("Jumlah node dan edge di Twitter:", twitter_graph.vcount(), twitter_graph.ecount())
```

CENTRALLITY MEASURES

```
def calculate_network_metrics(graph):
    n = len(graph.vs)
    max_possible_betweenness = (n - 1) * (n - 2) / 2
    betweenness_centrality = graph.betweenness()
    betweenness_centrality_normalized = [b / max_possible_betweenness for b in betweenness_centrality]
    avg_betweenness_centrality_normalized = sum(betweenness_centrality_normalized) / n
    closeness_centrality = graph.closeness()
    avg_closeness_centrality = sum(closeness_centrality) / n
    degree_centrality = graph.degree()
    degree_centrality_normalized = [d / (n - 1) for d in degree_centrality]
    avg_degree_centrality_normalized = sum(degree_centrality_normalized) / n
    clustering_coefficient = graph.transitivity_local_undirected()
    global_clustering_coefficient = graph.transitivity_undirected()
    def local_efficiency(graph):
        local_eff = []
        for v in range(len(graph.vs)):
            neighbors = graph.neighbors(v)
            if len(neighbors) < 2:
                local_eff.append(0)
            else:
                subgraph = graph.subgraph(neighbors)
                local_eff.append(subgraph.transitivity_undirected() or 0)
        return local_eff
    local_efficiency_values = local_efficiency(graph)
    avg_local_efficiency = sum(local_efficiency_values) / n
    return {
        'betweenness_centrality': betweenness_centrality_normalized,
        'avg_betweenness_centrality': avg_betweenness_centrality_normalized,
        'closeness_centrality': closeness_centrality,
        'avg_closeness_centrality': avg_closeness_centrality,
        'degree_centrality': degree_centrality_normalized,
        'avg_degree_centrality': avg_degree_centrality_normalized,
        'clustering_coefficient': clustering_coefficient,
        'global_clustering_coefficient': global_clustering_coefficient,
        'local_efficiency': local_efficiency_values,
        'avg_local_efficiency': avg_local_efficiency
    }
twitter_metrics = calculate_network_metrics(twitter_graph)
instagram_metrics = calculate_network_metrics(instagram_graph)
facebook_metrics = calculate_network_metrics(facebook_graph)
print("Degree Centrality:", twitter_metrics['degree_centrality'])
print("Rata-rata Degree Centrality:", twitter_metrics['avg_degree_centrality'])
print("Betweenness Centrality:", twitter_metrics['betweenness_centrality'])
print("Rata-rata Betweenness Centrality:", twitter_metrics['avg_betweenness_centrality'])
```

```
print("Closeness Centrality:", twitter_metrics['closeness_centrality'])
print("Rata-rata Closeness Centrality:", twitter_metrics['avg_closeness_centrality'])
print("Clustering Coefficient:", twitter_metrics['clustering_coefficient'])
print("Global Clustering Coefficient:", twitter_metrics['global_clustering_coefficient'])
print("Local Efficiency:", twitter_metrics['local_efficiency'])
print("Rata-rata Local Efficiency:", twitter_metrics['avg_local_efficiency'])
print("Degree Centrality Instagram:", instagram_metrics['degree_centrality'])
print("Rata-rata Degree Centrality Instagram:", instagram_metrics['avg_degree_centrality'])
print("Betweenness Centrality Instagram:", instagram_metrics['betweenness_centrality'])
print("Rata-rata Betweenness Centrality Instagram:", instagram_metrics['avg_betweenness_centrality'])
print("Closeness Centrality Instagram:", instagram_metrics['closeness_centrality'])
print("Rata-rata Closeness Centrality Instagram:", instagram_metrics['avg_closeness_centrality'])
print("Clustering Coefficient Instagram:", instagram_metrics['clustering_coefficient'])
print("Global Clustering Coefficient Instagram:", instagram_metrics['global_clustering_coefficient'])
print("Local Efficiency Instagram:", instagram_metrics['local_efficiency'])
print("Rata-rata Local Efficiency Instagram:", instagram_metrics['avg_local_efficiency'])
print("Degree Centrality Facebook:", facebook_metrics['degree_centrality'])
print("Rata-rata Degree Centrality Facebook:", facebook_metrics['avg_degree_centrality'])
print("Betweenness Centrality Facebook:", facebook_metrics['betweenness_centrality'])
print("Rata-rata Betweenness Centrality Facebook:", facebook_metrics['avg_betweenness_centrality'])
print("Closeness Centrality Facebook:", facebook_metrics['closeness_centrality'])
print("Rata-rata Closeness Centrality Facebook:", facebook_metrics['avg_closeness_centrality'])
print("Clustering Coefficient Facebook:", facebook_metrics['clustering_coefficient'])
print("Global Clustering Coefficient Facebook:", facebook_metrics['global_clustering_coefficient'])
print("Local Efficiency Facebook:", facebook_metrics['local_efficiency'])
print("Rata-rata Local Efficiency Facebook:", facebook_metrics['avg_local_efficiency'])
```

```
clustering_facebook_walktrap = facebook_graph_100.community_walktrap()
clusters_facebook_walktrap = clustering_facebook_walktrap.as_clustering()
labels_facebook_walktrap = clusters_facebook_walktrap.membership
palette_facebook_walktrap = ig.RainbowPalette(len(set(labels_facebook_walktrap)))
num_communities_facebook_walktrap = len(set(labels_facebook_walktrap))
modularity_facebook_walktrap = clusters_facebook_walktrap.modularity
community_sizes_facebook_walktrap = [labels_facebook_walktrap.count(i) for i in set(labels_facebook_walktrap)]
average_members_facebook_walktrap = sum(community_sizes_facebook_walktrap) / len(community_sizes_facebook_walktrap)
print(f"Jumlah Cluster Facebook (Walktrap): {num_communities_facebook_walktrap}")
print(f"Modularitas Facebook (Walktrap): {modularity_facebook_walktrap}")
print("Anggota Cluster per Grup Facebook (Walktrap):", community_sizes_facebook_walktrap)
ig.plot(
    clusters_facebook_walktrap,
    vertex_label=facebook.iloc[:100, 0].tolist(),
    palette=palette_facebook_walktrap,
    vertex_size=20,
    layout=facebook_graph_100.layout("fr")
)
```

COMMUNITY DETECTION

```
twitter_numpy_100 = twitter_numpy[:100, :100]
undirected_matrix_twitter_100 = np.maximum(twitter_numpy_100, twitter_numpy_100.T)
twitter_graph_100 = ig.Graph.Adjacency(undirected_matrix_twitter_100.astype(bool).tolist(), mode=ig.ADJ_UNDIRECTED)
print(f"Jumlah node dalam graph: {twitter_graph_100.vcount()}")
community_twitter = twitter_graph_100.community_label_propagation()
print(f"Jumlah Komunitas Twitter: {len(set(community_twitter.membership))}")
print("Anggota Komunitas per Grup Twitter:", [list(community_twitter.membership).count(i) for i in set(community_twitter.membership)])
vertex_labels_twitter = twitter.iloc[:100, 0].tolist()
ig.plot(
    community_twitter,
    vertex_label=vertex_labels_twitter,
    palette=ig.RainbowPalette(len(set(community_twitter.membership))),
    vertex_size=20,
    layout=twitter_graph_100.layout("fr")
)
instagram_numpy_100 = instagram_numpy[:100, :100]
```

```
undirected_matrix_instagram_100 = np.maximum(instagram_numpy_100, instagram_numpy_100.T)
instagram_graph_100 = ig.Graph.Adjacency(undirected_matrix_instagram_100.astype(bool).tolist(), mode=ig.ADJ_UNDIRECTED)
print(f"Jumlah node dalam graph: {instagram_graph_100.vcount()}")
community_instagram = instagram_graph_100.community_label_propagation()
print(f"Jumlah Komunitas Instagram: {len(community_instagram)}")
print("Anggota Komunitas per Grup Instagram:", [len(c) for c in community_instagram])
vertex_labels_instagram = instagram.iloc[:100, 0].tolist()
ig.plot(
    community_instagram,
    vertex_label=vertex_labels_instagram,
    palette=ig.RainbowPalette(len(community_instagram)),
    vertex_size=20,
    layout=instagram_graph_100.layout("fr")
)
facebook_numpy_100 = facebook_numpy[:100, :100]
undirected_matrix_facebook_100 = np.maximum(facebook_numpy_100, facebook_numpy_100.T)
facebook_graph_100 = ig.Graph.Adjacency(undirected_matrix_facebook_100.astype(bool).tolist(), mode=ig.ADJ_UNDIRECTED)
print(f"Jumlah node dalam graph: {facebook_graph_100.vcount()}")
community_facebook = facebook_graph_100.community_label_propagation()
print(f"Jumlah Komunitas Facebook: {len(community_facebook)}")
print("Anggota Komunitas per Grup Facebook:", [len(c) for c in community_facebook])
vertex_labels_facebook = facebook.iloc[:100, 0].tolist()
ig.plot(
    community_facebook,
    vertex_label=vertex_labels_facebook,
    palette=ig.RainbowPalette(len(community_facebook)),
    vertex_size=20,
    layout=facebook_graph_100.layout("fr")
)
```

CLUSTERING

```
clustering_twitter_walktrap = twitter_graph_100.community_walktrap()
clusters_twitter_walktrap = clustering_twitter_walktrap.as_clustering()
labels_twitter_walktrap = clusters_twitter_walktrap.membership
palette_twitter_walktrap = ig.RainbowPalette(len(set(labels_twitter_walktrap)))
num_communities_twitter_walktrap = len(set(labels_twitter_walktrap))
modularity_twitter_walktrap = clusters_twitter_walktrap.modularity
community_sizes_twitter_walktrap = [labels_twitter_walktrap.count(i) for i in set(labels_twitter_walktrap)]
average_members_twitter_walktrap = sum(community_sizes_twitter_walktrap) / len(community_sizes_twitter_walktrap)
print(f"Jumlah Cluster Twitter (Walktrap): {num_communities_twitter_walktrap}")
print(f"Modularitas Twitter (Walktrap): {modularity_twitter_walktrap}")
print("Anggota Cluster per Grup Twitter (Walktrap):", community_sizes_twitter_walktrap)
ig.plot(
    clusters_twitter_walktrap,
    vertex_label=twitter.iloc[:100, 0].tolist(),
    palette=palette_twitter_walktrap,
    vertex_size=20,
    layout=twitter_graph_100.layout("fr")
)
clustering_instagram_walktrap = instagram_graph_100.community_walktrap()
clusters_instagram_walktrap = clustering_instagram_walktrap.as_clustering()
labels_instagram_walktrap = clusters_instagram_walktrap.membership
palette_instagram_walktrap = ig.RainbowPalette(len(set(labels_instagram_walktrap)))
num_communities_instagram_walktrap = len(set(labels_instagram_walktrap))
modularity_instagram_walktrap = clusters_instagram_walktrap.modularity
community_sizes_instagram_walktrap = [labels_instagram_walktrap.count(i) for i in set(labels_instagram_walktrap)]
average_members_instagram_walktrap = sum(community_sizes_instagram_walktrap) / len(community_sizes_instagram_walktrap)
print(f"Jumlah Cluster Instagram (Walktrap): {num_communities_instagram_walktrap}")
print(f"Modularitas Instagram (Walktrap): {modularity_instagram_walktrap}")
print("Anggota Cluster per Grup Instagram (Walktrap):", community_sizes_instagram_walktrap)
ig.plot(
    clusters_instagram_walktrap,
    vertex_label=instagram.iloc[:100, 0].tolist(),
    palette=palette_instagram_walktrap,
    vertex_size=20,
    layout=instagram_graph_100.layout("fr")
)
```