Social Network Model on Instagram user using Gephy with Python

1301190318_Iqbal Saviola Syah B, 1301194039_Farel Edris Putra

Abstract—The decision to use Instagram data for this project was motivated by several factors. Instagram is a leading social media platform globally, with a large user base and diverse interactions daily, providing a rich dataset for exploration. Its dynamic nature reflects contemporary social dynamics, making it a relevant example for studying user behaviors, relationships, and content consumption patterns.

Using social network analysis (SNA) as the modeling framework allows for a deep dive into Instagram user behavior, in this case, writers following users are used. Tools like Gephi enable the analysis of the social network, identification of key users, and exploration of patterns of influence and connectivity. This analytical approach facilitates the visualization and quantification of individual user centrality within the network.

The primary goal of the project is to understand comprehensively to network analysis and centrality methods through a real-world example from Instagram. By collecting social data from the platform, the project aims to provide great understanability with practical skills in data acquisition and preprocessing using Python, and graph visualization using Gephy for further analysis.

The dataset for the project is sourced from Instagram, focusing on aspects such as user interactions, following networks, and data processing of content metadata. The raw data is then processed using Python, including parsing, cleaning, and formatting, to prepare it for network analysis.

Index Terms—instagram, python, gephy, centrality, data preprocessing

I. INTRODUCTION

The decision to utilize Instagram data for this project stems from several compelling reasons. Firstly, Instagram stands as one of the most prominent social media platforms globally, boasting a vast user base and a myriad of interactions daily. Its dynamic nature provides a rich dataset ripe for exploration, encompassing diverse user behaviors, relationships, and content consumption patterns. By leveraging Instagram data, participants gain access to a real-world example that resonates with contemporary social dynamics, offering insights applicable across various domains.

Utilizing social network analysis (SNA) as our modeling framework allows us to dissect and understand the structure of data and its user behavior. Using Gephi, its provides us to analyze the structure of the social network, identify key actors (users), and uncover patterns of influence and connectivity. Through this analytical lens, we can visualize the network, and quantify the centrality of individual users.

II. RELATED WORK

We collected Instagram data using Python and constructed a social network using Gephi. We then analyzed the network to identify key influencers, communities, and patterns of interaction among users.

Our analysis revealed several interesting findings, including the presence of distinct communities within the Instagram network and the identification of influential users who play a significant role in information dissemination.

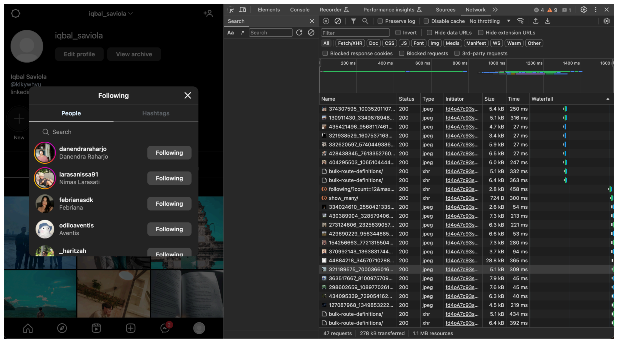
III. METHOD AND EXPERIMENTS

A. Instagram Data Collection and Preprocessing

Social media data in general are unique in their data structure, such as Instagram in this case. Also to issue of legal activity of scraping Meta / Facebook / Instagram data by choice is considered illegal, which we define legal way to extract it. By processes, we determine to divide into three chapter ;

1) *Instagram data collection*

The method we used to collect the data from Instagram, is called Network Traffic. In another word, we call the API to perform the collection data for us, in this case we were interested in extracting following data, and profile info. In order to do that, we use devtools in a web browser to save the network traffic, which includes the API calls and responses, as a Har file. This file can then be easily passed into Python for processing.



Name	Status	Type	Initiator	Size	Time	Waterfall
374307595_10035201107...	200	jpeg	fd4oA7c93s...	5.4 kB	250 ms	
130911430_3349878948...	200	jpeg	fd4oA7c93s...	5.1 kB	316 ms	
435421496_9568117461...	200	jpeg	fd4oA7c93s...	4.7 kB	27 ms	
321938529_1607537163...	200	jpeg	fd4oA7c93s...	3.4 kB	27 ms	
332620597_5740449386...	200	jpeg	fd4oA7c93s...	5.9 kB	27 ms	
428438345_7613352760...	200	jpeg	fd4oA7c93s...	6.5 kB	27 ms	
404295503_1065104444...	200	jpeg	fd4oA7c93s...	6.0 kB	247 ms	
bulk-route-definitions/	200	xhr	fd4oA7c93s...	5.1 kB	332 ms	
bulk-route-definitions/	200	xhr	fd4oA7c93s...	6.4 kB	363 ms	
following/?count=12&max...	200	xhr	fd4oA7c93s...	2.8 kB	458 ms	
show_many/	200	xhr	fd4oA7c93s...	724 B	300 ms	
334024610_2550421335...	200	jpeg	fd4oA7c93s...	2.6 kB	54 ms	
430389904_328579406...	200	jpeg	fd4oA7c93s...	7.3 kB	213 ms	
273124606_2325639057...	200	jpeg	fd4oA7c93s...	6.3 kB	221 ms	
429690229_956344885...	200	jpeg	fd4oA7c93s...	6.6 kB	53 ms	
154256663_7721315504...	200	jpeg	fd4oA7c93s...	7.3 kB	280 ms	
370992143_1363831744...	200	jpeg	fd4oA7c93s...	3.7 kB	94 ms	
44884218_34570710288...	200	jpeg	fd4oA7c93s...	28.8 kB	365 ms	
321189575_7000366016...	200	jpeg	fd4oA7c93s...	5.1 kB	309 ms	
363517667_8100975709...	200	jpeg	fd4oA7c93s...	7.9 kB	45 ms	
298602659_1089770261...	200	jpeg	fd4oA7c93s...	7.6 kB	45 ms	
434095339_729054162...	200	jpeg	fd4oA7c93s...	6.3 kB	40 ms	
127087968_1349853222...	200	jpeg	fd4oA7c93s...	4.5 kB	219 ms	
bulk-route-definitions/	200	xhr	fd4oA7c93s...	5.1 kB	434 ms	
bulk-route-definitions/	200	xhr	fd4oA7c93s...	6.4 kB	392 ms	

47 requests | 278 kB transferred | 1.1 MB resources

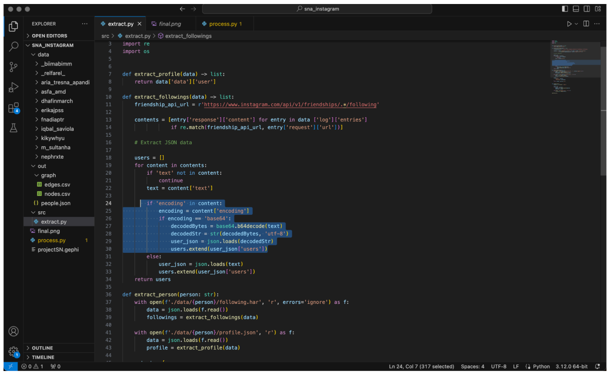
Figure 1: Extracting following data to Har File

After extracting Har file for following data and json for profile data, we process the data in Python and store it in a format that Gephi can understand before loading it into Gephi for visualization and exploration.

Keep in mind, that we determine to limit the data collection to our friends' accounts and limit ourself to those with a small number of followers to avoid any potential legal issues.

2) *Disassemble HAR files*

After extracting the data from instagram API, we focused on processing and preparing the data for Gephi by examining HTTP archives and extracting user information from requests and responses. They demonstrate how to check for specific URLs and use regular expressions to match and extract user data. The main issue we found during this process, is handling base64-encoded data in the response data to extract the user information. To handle this, we use encoding the base64-encoded data to extract the user information.

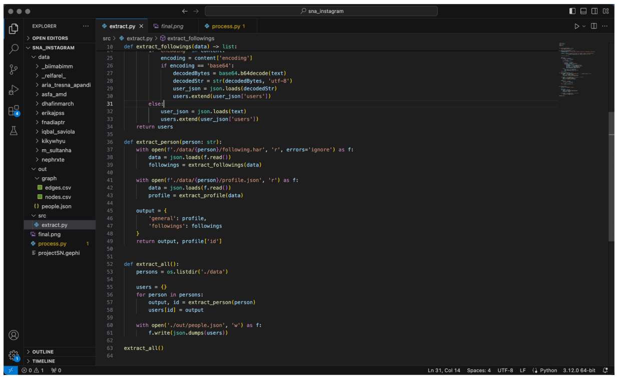


```python
import re
import os


def extract_profile(data) -> list:
    return data['data']['user']

def extract_followings(data) -> list:
    friendship_api_url = r'https://www.instagram.com/api/v1/friendships/.*/following'

    contents = [entry['response']['content'] for entry in data ['log']['entries']
                if re.match(friendship_api_url, entry['request']['url'])]

    # Extract JSON data

    users = []
    for content in contents:
        if 'text' not in content:
            continue
        text = content['text']

        if 'encoding' in content:
            encoding = content['encoding']
            if encoding == 'base64':
                decodedBytes = base64.b64decode(text)
                decodedStr = str(decodedBytes, 'utf-8')
                user_json = json.loads(decodedStr)
                users.extend(user_json['users'])
        else:
            user_json = json.loads(text)
            users.extend(user_json['users'])
    return users

def extract_person(person: str):
    with open(f'./data/{person}/following.har', 'r', errors='ignore') as f:
        data = json.loads(f.read())
        followings = extract_followings(data)

    with open(f'./data/{person}/profile.json', 'r') as f:
        data = json.loads(f.read())
        profile = extract_profile(data)
```

Figure 2: extract.py, encoding base-64 process

After that, we also need to export the user data to another JSON file we named "people.json." To automate the process, we write a function called "extract_followings" that takes data as an argument and returns a list of user objects. The function is then called to extract all users by iterating over a list of usernames and calling the "extract_person" function for each user.

After that, we need to create a dictionary instead of a list to store user data, which includes the user ID and the list of users that person is following. The function to extract user profile data is also introduced, which returns the entire user object from the "profile.json" file. After extracting user-profiles and their followings, the data is combined into an object and written to a JSON file named "people.json."

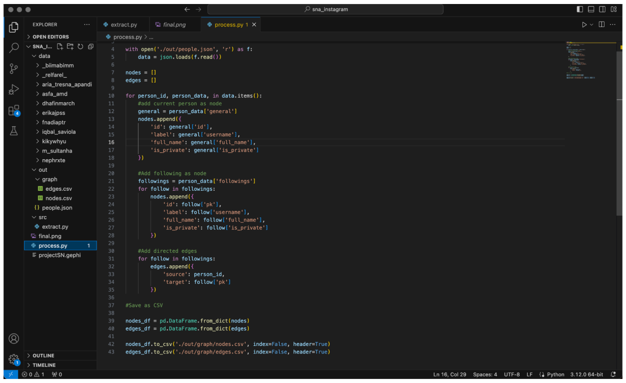


```python
def extract_followings(data) -> list:
            encoding = content['encoding']
            if encoding == 'base64':
                decodedBytes = base64.b64decode(text)
                decodedStr = str(decodedBytes, 'utf-8')
                user_json = json.loads(decodedStr)
                users.extend(user_json['users'])
        else:
            user_json = json.loads(text)
            users.extend(user_json['users'])
    return users

def extract_person(person: str):
    with open(f'./data/{person}/following.har', 'r', errors='ignore') as f:
        data = json.loads(f.read())
        followings = extract_followings(data)

    with open(f'./data/{person}/profile.json', 'r') as f:
        data = json.loads(f.read())
        profile = extract_profile(data)

    output = {
        'general': profile,
        'followings': followings
    }
    return output, profile['id']


def extract_all():
    persons = os.listdir('./data')

    users = {}
    for person in persons:
        output, id = extract_person(person)
        users[id] = output

    with open('./out/people.json', 'w') as f:
        f.write(json.dumps(users))

extract_all()
```

Figure 3: extract.py, extract person & all function

3) *Process data - Prepare for Gephi (nodes & edges)*

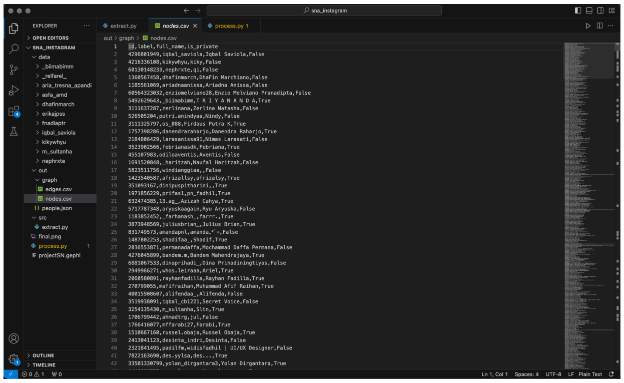
After we extract and disassemble the HAR files, we prepare and transform the previous JSON file into two CSV files, one for nodes and one for edges, using Python dictionaries. The nodes CSV file includes unique identifiers and labels, while the edges file connects nodes with source and target identifiers. To create a node, an ID and a label (username) are required, along with other attributes such as full name and privacy status. The edges are directed, with the source being the current user and the target being the user being followed. The processed data is saved as CSV files, which can be imported into Gephi for visualization.



```python
with open('./out/people.json', 'r') as f:
    data = json.loads(f.read())

nodes = []
edges = []

for person_id, person_data, in data.items():
    #add current person as node
    general = person_data['general']
    nodes.append({
        'id': general['id'],
        'label': general['username'],
        'full_name': general['full_name'],
        'is_private': general['is_private']
    })

    #Add following as node
    followings = person_data['followings']
    for follow in followings:
        nodes.append({
            'id': follow['pk'],
            'label': follow['username'],
            'full_name': follow['full_name'],
            'is_private': follow['is_private']
        })

    #Add directed edges
    for follow in followings:
        edges.append({
            'source': person_id,
            'target': follow['pk']
        })

#Save as CSV

nodes_df = pd.DataFrame.from_dict(nodes)
edges_df = pd.DataFrame.from_dict(edges)

nodes_df.to_csv('./out/graph/nodes.csv', index=False, header=True)
edges_df.to_csv('./out/graph/edges.csv', index=False, header=True)
```

Figure 4: process.py, extract person & all function

We also emphasize the importance of separating the data extraction and processing into two stages, as the extracted data can be used for various purposes beyond just importing into Gephi.



```
id,label,full_name,is_private
4296801949,iqbal_saviola,Iqbal Saviola,False
4216336108,kikywhyu,kiky,False
60130148233,nephrxte,qi,False
1360567458,dhafinmarch,DhaFin Marchiano,False
1185561069,ariadnaanissa,Ariadna Anissa,False
60564323032,enziomelviano28,Enzio Melviano Pranadipta,False
5492629643,_biimabimm,T R I Y A N A N D A,True
3111637287,zerlinana,Zerlina Natasha,False
526505204,putri.anindyaa,Nindy,False
3111325797,os_088,Firdaus Putra K,True
1757398286,danendraraharjo,Danendra Raharjo,True
2184806429,larasanissa91,Nimas Larasati,False
3523902566,febrianasdk,Febriana,True
455107983,odiloaventis,Aventis,False
1691520848,_haritzah,Naufal Haritzah,False
5823511756,windianggiaa,,False
1423540587,afrizallsy,afrizalsy,True
351093167,dinipuspitharini,,True
1971856229,prifas1,pn_fadhil,True
632474385,13.ag_,Azizah Cahya,True
5717787348,aryuskaagain,Ryu Aryuska,False
1183852452,_farhanash_,farrr.,True
3873948569,juliusbrian_,Julius Brian,True
831749573,amandapnl,amanda｡*ﾟ+,False
1487802253,shadifaa_,Shadif,True
2036553871,permanadaffa,Mochammad Daffa Permana,False
4276045899,bandem.m,Bandem Mahendrajaya,True
6881067533,dinaprihadi_,Dina Prihadiningtiyas,False
2949966271,whos.leiraaa,Ariel,True
2060580891,rayhanfadilla,Rayhan Fadilla,True
270799055,mafifraihan,Muhammad Afif Raihan,True
40015908607,alifendaa_,Alifenda,False
3519938091,iqbal_cb1221,Secret Voice,False
3254135430,m_sultanha,Sltn,True
1706799442,ahmadtrg,jul,False
1766416077,mffarabi27,Farabi,True
1510667160,russel.obaja,Russel Obaja,True
2413041123,desinta_indri,Desinta,False
2321841495,padilfm,widisfadhil | UI/UX Designer,False
7822163690,des.yylsa,des...,True
33501330799,yolan_dirgantara3,Yolan Dirgantara,True
```

Figure 5: Final nodes data, ready for Gephy importing

B. Formula and Calculating of Various Centrality

Within this case, we focused only on using the 3 Centrality method, Betweenness Centrality, Closeness Centrality, and Page Rank. Those 4 methods are commonly used to determine the centrality of the Social Network Model, in this case, we wanted to understand which one is the accurate method to determine the centrality of the Instagram Network model.

1) *Betweenness Centrality*

Betweenness centrality is a metric that measures how central a node is in a graph based on the shortest paths between other edges. This method widely used measure for determining how much a node contributes to information passing from one part of a network to another. The betweenness centrality of a node v is given by the expression (1).

$$g(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}} \quad (1)$$

A good measure of the centrality of a node is when the nodes incorporate more global information at any two given nodes in the network [1].

2) *Closeness Centrality*

Closeness centrality is a measure of how close a node is to all other nodes in a network. It is calculated as the average of the shortest path length from the node to every other node in the network. Closeness centrality is defined as;

$$C_c(i) = \left[\sum_{j=1}^{N} d(i,j)\right]^{-1}$$

$$C_C^{'}(i) = (C_C(i))/(N-1) \quad (2)$$

Where defined in (2), Distance d(i,j) between vertices i and j is the length of the shortest i – j path.

3) *Page Rank*

PageRank (PR) is an algorithm used by Google Search to rank web pages in their search engine results. PageRank is a link analysis algorithm and it assigns a numerical weighting to each element of a hyperlinked set of documents, such as the World Wide Web, with the purpose of "measuring" its relative importance within the set. The PageRank is given by the expression (3).

$$x_i \propto \sum_{(j,i)\in E} x_j \quad (3)$$

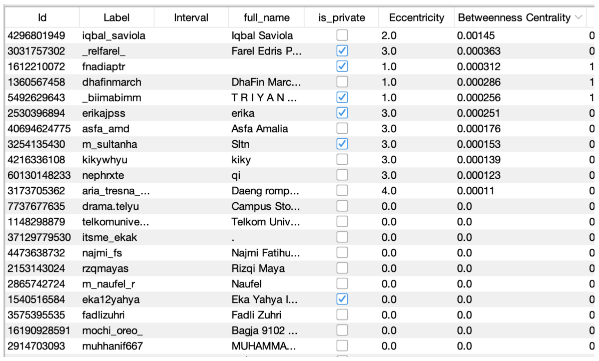
Where i's Rank score xi is the sum of the Rank scores xj of all pages j that point to i.

C. Analyze the Centrality Value

As the method defined above (Betweenness Centrality, Closeness Centrality, and Page Rank) thereby the result Centrality value and its analysis;

1) *Betweenness Centrality Value*

Refer to figure 6, explained that the top 3 nodes that have the biggest value of Betweenness Centrality are node "iqbal_saviola", "_relfarel_", and "fnadiaptr". This result fundamentally affected by the centrality method, in which Betweenness Centrality behavior measures how central a node is in a graph based on the shortest paths between others. Therefore, the node "iqbal_saviola" node is having the shortest paths to all important node such as "_relfarel_" and "fnadiaptr".

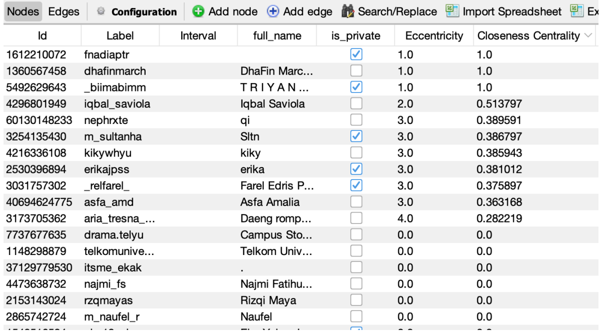


Id	Label	Interval	full_name	is_private	Eccentricity	Betweenness Centrality	
4296801949	iqbal_saviola		Iqbal Saviola	□	2.0	0.00145	0
3031757302	_relfarel_		Farel Edris P...	☑	3.0	0.000363	0
1612210072	fnadiaptr			☑	1.0	0.000312	1
1360567458	dhafinmarch		DhaFin Marc...	□	1.0	0.000286	1
5492629643	_biimabimm		T R I Y A N ...	☑	1.0	0.000256	1
2530396894	erikajpss		erika	☑	3.0	0.000251	0
40694624775	asfa_amd		Asfa Amalia	□	3.0	0.000176	0
3254135430	m_sultanha		Sltn	☑	3.0	0.000153	0
4216336108	kikywhyu		kiky	□	3.0	0.000139	0
60130148233	nephrxte		qi	□	3.0	0.000123	0
3173705362	aria_tresna_...		Daeng romp...	□	4.0	0.00011	0
7737677635	drama.telyu		Campus Sto...	□	0.0	0.0	0
1148298879	telkomunive...		Telkom Univ...	□	0.0	0.0	0
37129779530	itsme_ekak		.	□	0.0	0.0	0
4473638732	najmi_fs		Najmi Fatihu...	□	0.0	0.0	0
2153143024	rzqmayas		Rizqi Maya	□	0.0	0.0	0
2865742724	m_naufel_r		Naufel	□	0.0	0.0	0
1540516584	eka12yahya		Eka Yahya I...	☑	0.0	0.0	0
3575395535	fadlizuhri		Fadli Zuhri	□	0.0	0.0	0
16190928591	mochi_oreo_		Bagja 9102 ...	□	0.0	0.0	0
2914703093	muhhanif667		MUHAMMA...	□	0.0	0.0	0

Figure 6: Result of Betweenness Centrality value

2) *Closeness Centrality Value*

Refer to figure 7 below, that the top 3 nodes that have the biggest value of Closeness Centrality are node "fnadiaptr", "dhafinmarch", and "_biimabimm". This result is caused by Closeness Centrality behavior that measure of how close a node is to all other nodes. Meaning, the more following / edges that node has, the more likely it becomes a centrality.

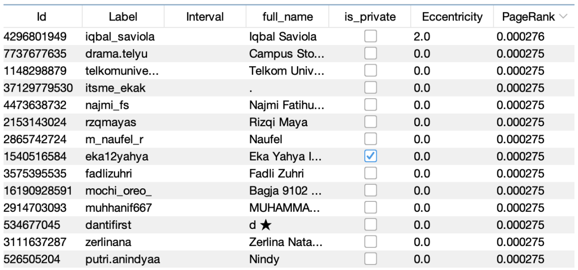


Id	Label	Interval	full_name	is_private	Eccentricity	Closeness Centrality
1612210072	fnadiaptr			☑	1.0	1.0
1360567458	dhafinmarch		DhaFin Marc...	□	1.0	1.0
5492629643	_biimabimm		T R I Y A N ...	☑	1.0	1.0
4296801949	iqbal_saviola		Iqbal Saviola	□	2.0	0.513797
60130148233	nephrxte		qi	□	3.0	0.389591
3254135430	m_sultanha		Sltn	☑	3.0	0.386797
4216336108	kikywhyu		kiky	□	3.0	0.385943
2530396894	erikajpss		erika	☑	3.0	0.381012
3031757302	_relfarel_		Farel Edris P...	☑	3.0	0.375897
40694624775	asfa_amd		Asfa Amalia	□	3.0	0.363168
3173705362	aria_tresna_...		Daeng romp...	□	4.0	0.282219
7737677635	drama.telyu		Campus Sto...	□	0.0	0.0
1148298879	telkomunive...		Telkom Univ...	□	0.0	0.0
37129779530	itsme_ekak		.	□	0.0	0.0
4473638732	najmi_fs		Najmi Fatihu...	□	0.0	0.0
2153143024	rzqmayas		Rizqi Maya	□	0.0	0.0
2865742724	m_naufel_r		Naufel	□	0.0	0.0

Figure 6: Result of Closeness Centrality value

3) *Page Rank Value*

Refer to Figure 7 below, that the top 3 nodes that have the biggest value of Page Rank are node "iqbal_saviola", "drana.telyu", and "telkomuniversity". This result is fundamentally affected by PageRank behavior, which works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. Page rank also have an underlying assumption, that more important websites (in this case node) are likely to receive more links (in this case edge) from other websites (nodes).



Id	Label	Interval	full_name	is_private	Eccentricity	PageRank
4296801949	iqbal_saviola		Iqbal Saviola	□	2.0	0.000276
7737677635	drama.telyu		Campus Sto...	□	0.0	0.000275
1148298879	telkomunive...		Telkom Univ...	□	0.0	0.000275
37129779530	itsme_ekak		.	□	0.0	0.000275
4473638732	najmi_fs		Najmi Fatihu...	□	0.0	0.000275
2153143024	rzqmayas		Rizqi Maya	□	0.0	0.000275
2865742724	m_naufel_r		Naufel	□	0.0	0.000275
1540516584	eka12yahya		Eka Yahya I...	☑	0.0	0.000275
3575395535	fadlizuhri		Fadli Zuhri	□	0.0	0.000275
16190928591	mochi_oreo_		Bagja 9102 ...	□	0.0	0.000275
2914703093	muhhanif667		MUHAMMA...	□	0.0	0.000275
534677045	dantifirst		d ★	□	0.0	0.000275
3111637287	zerlinana		Zerlina Nata...	□	0.0	0.000275
526505204	putri.anindyaa		Nindy	□	0.0	0.000275

Figure 7: Result of Page Rank value

Interestingly, all the value that page rank created, is similar to one another. The highest value is 2.76, and the lowest value is 2.74, with only 0.02 differences. Page Rank values have a low distribution of one another, which makes this method irrelevant.

D. Visualization of Result (Network)

The Social Network Model on Instagram below is representative of 11 main nodes (e.g. main account) which all correlated to the “iqbal_saviola” account. The Social Network model with Betweenness Centrality would look like this;

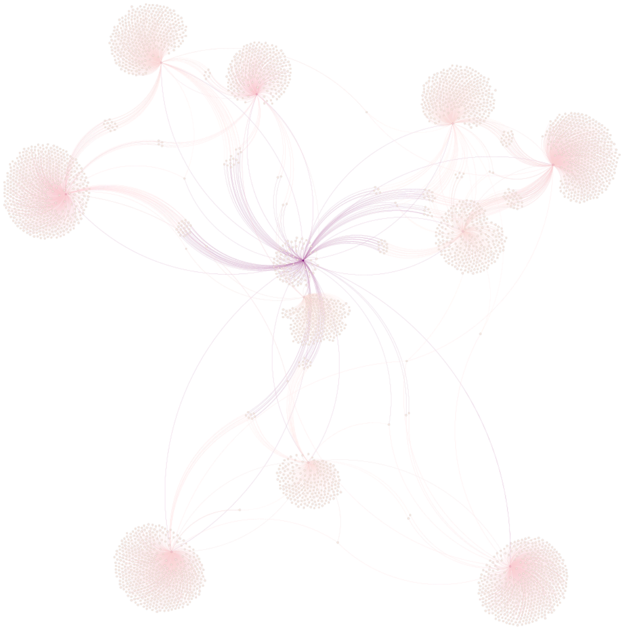


Figure 8: Visualization of Betweenness Centrality

The Figure above intuitively shows that the middle part which is highly colored Purple, is central to the network. The middle node is the node “iqbal_saviola”, which is highly colored purple and is central to the network. This is because the Betweenness Centrality is the shortest path between other nodes. Resulting relevant centrality, as we can intuitively see in Figure 8, node “iqbal_saviola” perfectly sits in the middle of the network.

Another result, when the Closeness Centrality method is being used, would look like Figure 9 below. The figure below show that the nodes that are highly colored Purple are “fnadiaptr” (upper left node) “_biimabimm” (upper right node), “dhafinmarch” (lower right) Those three nodes are becoming the central of the network, according to Betweenness Centrality. Resulting irrelevant centrality, as we can intuitively see in Figure 9, those 3 nodes doesn’t perfectly sit in the middle of the network, but rather scatter around the network.

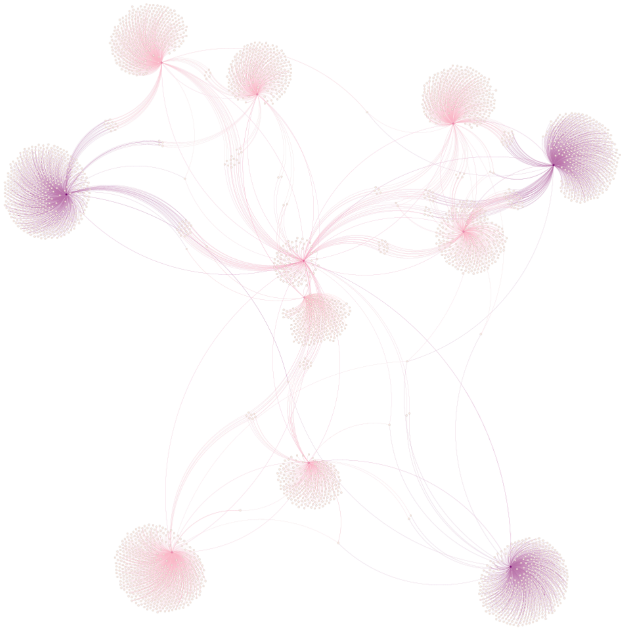


Figure 9: Visualization of Closeness Centrality

Another result, when the Closeness Centrality method is being used, would look like this;

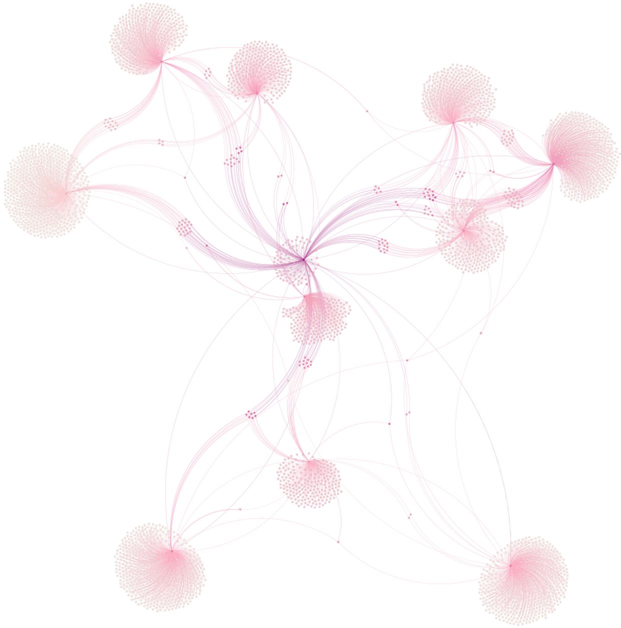


Figure 10: Visualization of Page Rank

The Figure above shows that the middle part which is highly Purple which the node “iqbal_saviola”, but yet we hardly identify the central of the network because the color doesn’t distinct from each other. It is caused by Page Rank that having a low distribution values of one another. Compared with 2 previous figures (Betweenness and Closeness centrality) are easier to identify, because the color easy to identify, which is pink and which is purple.

IV. CONCLUSION

By identifying the 3 Figures above, we can conclude that Betweenness Centrality is the most relevant method for this case, Instagram following data. It shows the most relevant centrality, the node “iqbal_saviola” which perfectly sits in the middle of the network. Fundamentally caused by how the writer collects the data, all 11 main nodes are basically connected to node “iqbal_saviola” as stated in the beginning, which his friends and colleagues used for this case study.

IV. REFERENCES

[1] J. U. Duncombe, “Betweenness centrality in large complex networks” *Springer*, vol. 38, pp. 163–168, 2004, doi: https://doi.org/10.1140/epjb/e2004-00111-4.