

Analisis Data Pemilu 2024 Dengan Mendeteksi Komunitas Menggunakan Metode Girvan Newman

Azhar Amirul Abid
Leonardus Adi Widjayanto

1301213032
1301213423

1

Abstract—This study aims to analyze the 2024 election data by detecting communities using the Girvan-Newman method. Data was obtained through a self-conducted crawling process from the X application, encompassing user interactions and conversations related to the election. The Girvan-Newman method was employed to identify community structures within the social network formed by voter and candidate interactions. By applying this method, the study successfully revealed interaction patterns and voter segmentation based on political preferences and demographics. The analysis results indicate significant communities that can help in understanding political dynamics and effective campaign strategies. This research provides important contributions to the field of social network analysis and political studies, and can serve as a reference for further research and strategic planning by political parties.

Keyword – Election Data Analysis, Community Detection, Girvan Detection, Social Network Analysis, Data Crawling

I. PENDAHULUAN

Pemilu 2024 di Indonesia merupakan peristiwa penting dalam demokrasi yang melibatkan jutaan pemilih dan berbagai kandidat. Analisis data pemilu menjadi krusial untuk memahami dinamika politik, preferensi pemilih, dan strategi kampanye yang efektif. Dalam era digital, media sosial seperti aplikasi X memainkan peran besar dalam interaksi politik dan penyebaran informasi.

Penelitian ini memanfaatkan data dari aplikasi X, yang diperoleh melalui proses crawling, untuk menganalisis percakapan dan interaksi terkait pemilu. Metode Girvan-Newman digunakan untuk mendeteksi komunitas dalam jaringan sosial yang terbentuk dari interaksi pemilih dan kandidat. Melalui metode ini, penelitian ini bertujuan mengungkap pola-pola interaksi dan segmentasi pemilih berdasarkan preferensi politik dan demografi. Hasil penelitian diharapkan memberikan kontribusi penting bagi analisis jaringan sosial dan studi politik, serta menjadi acuan bagi strategi kampanye yang lebih efektif.

II. DASAR TEORI

1. Analisis Jaringan Sosial (Social Network Analysis)

Analisis jaringan sosial adalah metodologi yang digunakan untuk memahami struktur hubungan sosial melalui grafis dan teori jaringan. Jaringan terdiri dari simpul (nodes) yang mewakili individu atau entitas, dan hubungan (edges) yang mewakili interaksi atau koneksi di antara mereka. Dalam konteks pemilu, analisis jaringan sosial digunakan untuk mempelajari bagaimana pemilih dan kandidat berinteraksi, serta bagaimana informasi dan pengaruh menyebar dalam jaringan tersebut. Analisis jaringan sosial juga memiliki dampak praktis yang signifikan seperti yang dibahas di [5] yang menyoroti komponen-komponen kritis dalam ekosistem jaringan komunitas untuk mengatasi kesenjangan digital yang persisten

2. Metode Girvan-Newman

Metode Girvan-Newman adalah salah satu algoritma untuk deteksi komunitas dalam jaringan sosial. Algoritma ini bekerja dengan menghapus tepi (edges) dengan nilai betweenness tertinggi, yang merupakan ukuran seberapa sering suatu tepi menjadi bagian dari jalur terpendek antar pasangan simpul dalam jaringan. Penghapusan tepi dengan betweenness tertinggi secara iteratif akan memecah jaringan menjadi komunitas-komunitas yang lebih kecil. Proses ini membantu mengidentifikasi struktur komunitas yang ada dalam jaringan, mengungkapkan kelompok-kelompok dengan interaksi yang lebih intens di dalamnya dibandingkan dengan luar kelompok.

3. Crawling Data

Crawling data adalah teknik untuk mengumpulkan informasi dari berbagai sumber online secara otomatis. Dalam penelitian ini, crawling data digunakan untuk mengumpulkan data dari aplikasi X. Proses ini melibatkan ekstraksi data dari percakapan dan interaksi pengguna terkait pemilu, yang kemudian dianalisis untuk mendeteksi komunitas dan pola interaksi. Untuk mendukung analisis ini, teknik crawling data menjadi krusial. Studi oleh Javed

et al. (2018) menawarkan tinjauan multidisiplin tentang pendekatan deteksi komunitas, menyoroti pentingnya data yang luas dan representatif untuk memvalidasi struktur komunitas yang teridentifikasi.

4. Pemilu dan Media Sosial

Dalam beberapa tahun terakhir, media sosial telah memainkan peran yang signifikan dalam pengaruh politik, terutama menjelang pemilihan umum (pemilu) [1]. Media sosial telah menjadi platform penting dalam konteks pemilu, di mana kandidat dan pemilih dapat berinteraksi secara langsung. Percakapan politik, kampanye, dan penyebaran informasi di media sosial memengaruhi opini publik dan perilaku pemilih. Studi tentang interaksi ini dapat memberikan wawasan penting tentang dinamika politik dan preferensi pemilih.

5. Segmentasi Politik

Segmentasi politik adalah proses mengelompokkan pemilih berdasarkan karakteristik tertentu, seperti preferensi politik, demografi, atau perilaku. Dengan memahami segmen-segmen ini, partai politik dapat merumuskan strategi kampanye yang lebih efektif dan tepat sasaran. Selain itu, kampanye politik dan partai politik telah melihat nilai strategis dari media sosial sebagai alat untuk mempengaruhi pemilih, menggalang dukungan, dan membangun citra yang diinginkan[2].

III. METODOLOGI PENELITIAN

A. Preprocessing data

1. Perhitungan Total Interaksi

- Baris Kode :

```
df['total_interaction'] = df['retweet_count'] + df['favorite_count']
df = df[['full_text', 'username', 'total_interaction']].sort_values(by='total_interaction', ascending=False)
df.head()
```

- Baris pertama (`df['total_interaction'] = df['retweet_count'] + df['favorite_count']`): Menambahkan kolom baru 'total_interaction' ke dataframe 'df'. Kolom ini adalah hasil penjumlahan antara kolom 'retweet_count' dan 'favorite_count', yang merepresentasikan total respons atau interaksi yang diterima oleh suatu konten dari pengguna.
- Baris kedua (`df = df[['full_text', 'username', 'total_interaction']].sort_values(by='total_interaction', ascending=False)`): Memilih kolom 'full_text', 'username', dan 'total_interaction' dari dataframe 'df'

dan mengurutkannya berdasarkan nilai 'total_interaction' secara menurun (descending). Hal ini bertujuan untuk menyoroti konten-konten yang paling banyak diinteraksi oleh pengguna, dengan yang paling tinggi di bagian atas dataframe.

- Hasil :

	full_text	username	total_interaction
1	Gugat Hasil Pilpres 2024, Anies-Imin Minta Pem...	CNNIndonesia	3669.0
0	Atas nama pasangan Prabowo Subianto dan Gibran...	prabowo	3385.0
4	Jokowi, satu2nya alasan PSI bisa eksis sekalig...	Miduk17	1291.0
5	Dengan pengumuman hasil pemilu tadi malam maka...	PartaiSocmed	1193.0
17	Komisi Pemilihan Umum (KPU) telah merampungkan...	kegblgnunfaedh	1111.0

Langkah ini penting untuk mengidentifikasi dan menyoroti konten yang paling populer atau berpengaruh dalam dataset, sehingga memudahkan analisis lebih lanjut atau pengambilan keputusan berdasarkan data interaksi yang diterima.

2. Ekstraksi Mentions dari Teks

- Baris Kode :

```
df['mention'] = df['full_text'].str.findall(r'@.*?(?=\s|$)')
df.head()
```

- `df['mention'] = df['full_text'].str.findall(r'@.*?(?=\s|$)')`: Menggunakan ekspresi regular (regex) untuk mencari dan mengekstraksi semua mention yang dimulai dengan karakter '@' dan berakhir sebelum spasi atau akhir teks ('\$'). Hasil ekstraksi disimpan dalam kolom baru 'mention', di mana setiap entri berupa daftar (list) username yang disebutkan dalam teks.
- `df.head()`: Menampilkan lima baris pertama dari dataframe 'df' setelah kolom 'mention' ditambahkan, sehingga Anda dapat melihat hasil ekstraksi tersebut dalam konteks dataset.
- Hasil :

	full_text	username	total_interaction	mention
1	Gugat Hasil Pilpres 2024, Anies-Imin Minta Pem...	CNNIndonesia	3669.0	[]
0	Atas nama pasangan Prabowo Subianto dan Gibran...	prabowo	3385.0	[]
4	Jokowi, satu2nya alasan PSI bisa eksis sekalig...	Miduk17	1291.0	[]
5	Dengan pengumuman hasil pemilu tadi malam maka...	PartaiSocmed	1193.0	[]
17	Komisi Pemilihan Umum (KPU) telah merampungkan...	kegblgnunfaedh	1111.0	[]

Langkah ini berguna untuk mempersiapkan data untuk analisis yang lebih mendalam terkait dengan interaksi atau pembicaraan

yang melibatkan pengguna tertentu yang disebutkan dalam teks, seperti dalam konteks analisis sosial atau jaringan.

3. Menghapus Kolom 'full_text' dari DataFrame

- Baris kode :

```
df.drop(['full_text'], inplace=True, axis=1)
df.head()
```

- `df.drop(['full_text'], inplace=True, axis=1)`: Menghapus kolom 'full_text' dari dataframe 'df'. Parameter `inplace=True` menandakan bahwa perubahan akan diterapkan secara langsung pada dataframe 'df' tanpa perlu membuat dataframe baru. Parameter `axis=1` menunjukkan bahwa yang dihapus adalah kolom, bukan baris.
- `df.head()`: Setelah kolom 'full_text' dihapus, digunakan `df.head()` untuk menampilkan lima baris pertama dari dataframe 'df'. Hal ini membantu untuk memverifikasi bahwa kolom 'full_text' telah dihapus dan melihat struktur dataframe setelah penghapusan kolom tersebut.
- Hasil :

	username	total_interaction	mention
1	CNNIndonesia	3669.0	[]
0	prabowo	3385.0	[]
4	Miduk17	1291.0	[]
5	PartaiSocmed	1193.0	[]
17	kegblgnunfaedh	1111.0	[]

Langkah ini berguna jika kolom 'full_text' tidak lagi diperlukan dalam analisis atau jika ekstraksi informasi penting telah selesai dan fokus pindah ke atribut atau kolom lain dalam dataframe.

4. Meledakkan Kolom 'mention'

- Baris kode :

```
df = df.explode('mention')
df.info()
```

- `df.explode('mention')`: Mengubah setiap baris dalam dataframe 'df' yang memiliki multiple values dalam kolom 'mention' (yang awalnya berupa daftar) menjadi beberapa baris baru. Setiap nilai dalam daftar 'mention' akan menjadi baris baru, sehingga pengguna yang disebutkan lebih dari satu kali

akan muncul dalam beberapa baris yang berbeda.

- `df.info()`: Setelah melakukan eksplorasi (`explode`), digunakan `df.info()` untuk menampilkan informasi tentang dataframe 'df'. Ini termasuk jumlah baris dan kolom, serta tipe data dari setiap kolom setelah transformasi.
- Hasil :

```
<class 'pandas.core.frame.DataFrame'>
Index: 3474 entries, 1 to 3001
Data columns (total 3 columns):
#   Column             Non-Null Count  Dtype  
---  -
0   username            3472 non-null   object  
1   total_interaction    3472 non-null   float64  
2   mention             1016 non-null   object  
dtypes: float64(1), object(2)
memory usage: 108.6+ KB
```

Langkah ini berguna untuk mengubah struktur data dari satu yang memiliki beberapa nilai dalam satu baris menjadi banyak baris dengan satu nilai di kolom 'mention'. Hal ini memudahkan analisis lebih lanjut seperti pencarian pola atau hubungan antara pengguna yang disebutkan dan konten yang mereka sebutkan.

5. Mengubah Huruf Besar menjadi Huruf Kecil dalam Kolom 'mention'

- Baris Kode :

```
df['mention'] = df['mention'].str.lower()
df.head()
```

- `df['mention'].str.lower()`: Memanfaatkan metode `.str.lower()` pada kolom 'mention' dalam dataframe 'df'. Ini mengubah semua karakter dalam setiap nilai di kolom 'mention' menjadi huruf kecil.
- `df.head()`: Setelah transformasi dilakukan, digunakan `df.head()` untuk menampilkan lima baris pertama dari dataframe 'df'. Hal ini membantu untuk memverifikasi bahwa transformasi ke huruf kecil telah berhasil diterapkan pada kolom 'mention'.
- Hasil :

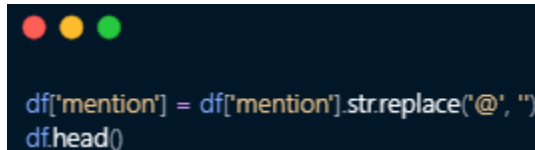
	username	total_interaction	mention
1	CNNIndonesia	3669.0	NaN
0	prabowo	3385.0	NaN
4	Miduk17	1291.0	NaN
5	PartaiSocmed	1193.0	NaN
17	kegblgnunfaedh	1111.0	NaN

Langkah ini bermanfaat untuk konsistensi dalam pengolahan teks, karena dalam banyak kasus

analisis teks, perbedaan huruf besar dan kecil tidak relevan dan dapat diabaikan. Transformasi ini memastikan bahwa data tetap konsisten dan siap untuk analisis lebih lanjut.

6. Menghapus Karakter '@' dari kolom 'mention'

- Baris kode :



```
df['mention'] = df['mention'].str.replace('@', '')
df.head()
```

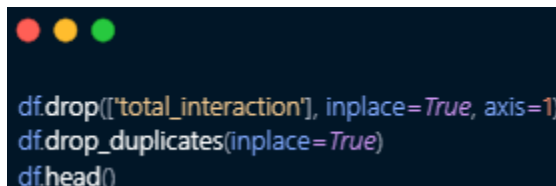
- `df['mention'].str.replace('@', '')`: Memanfaatkan metode `.str.replace()` pada kolom 'mention' dalam dataframe 'df'. Ini menggantikan setiap kemunculan karakter '@' dengan string kosong (''), efektif menghapus karakter '@' dari setiap nilai dalam kolom 'mention'.
- `df.head()`: Setelah transformasi dilakukan, digunakan `df.head()` untuk menampilkan lima baris pertama dari dataframe 'df'. Hal ini membantu untuk memverifikasi bahwa transformasi penghapusan karakter '@' telah berhasil diterapkan pada kolom 'mention'.
- Hasil :

	username	total_interaction	mention
1	CNNIndonesia	3669.0	NaN
0	prabowo	3385.0	NaN
4	Miduk17	1291.0	NaN
5	PartaiSocmed	1193.0	NaN
17	kegblgnunfaedh	1111.0	NaN

Langkah ini bermanfaat untuk membersihkan data dari karakter khusus yang tidak diperlukan atau mengganggu dalam teks, sehingga mempersiapkan data untuk analisis atau pemrosesan lanjutan dengan lebih baik.

7. Penghapusan Kolom 'total_interaction' dan baris duplikat

- Baris Kode :



```
df.drop(['total_interaction'], inplace=True, axis=1)
df.drop_duplicates(inplace=True)
df.head()
```

- `df.drop(['total_interaction'], inplace=True, axis=1)`: Menggunakan metode `.drop()` untuk menghapus kolom 'total_interaction' dari dataframe 'df'.

Parameter `inplace=True` menandakan bahwa perubahan akan diterapkan langsung pada dataframe 'df' tanpa membuat dataframe baru. Parameter `axis=1` menunjukkan bahwa yang dihapus adalah kolom

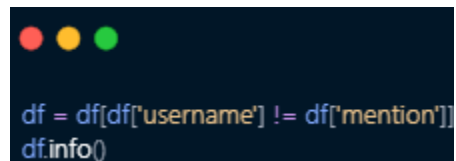
- `df.drop_duplicates(inplace=True)`: Menggunakan metode `.drop_duplicates()` untuk menghapus baris-baris yang memiliki nilai yang sama untuk semua kolomnya dari dataframe 'df'. Parameter `inplace=True` menjadikan perubahan ini diterapkan langsung pada dataframe 'df' tanpa membuat dataframe baru.
- Hasil :

	username	mention
3	gilang_ahm31272	officialmkri
8	MurtadhaOne1	ganjarpranowo
8	MurtadhaOne1	mohmahfudmd
10	Metro_TV	mohmahfudmd
10	Metro_TV	officialmkri

Kombinasi dari kedua operasi ini membantu mempersiapkan dataframe untuk analisis lebih lanjut dengan memastikan data bersih dari duplikat dan kolom yang tidak lagi diperlukan.

8. Menyaring Baris dengan Kondisi Tertentu

- Baris Kode :



```
df = df[df['username'] != df['mention']]
df.info()
```

- `df['username'] != df['mention']`: Ini adalah kondisi pembandingan antara nilai dalam kolom 'username' dan 'mention'. Baris ini akan menghasilkan sebuah series boolean yang bernilai True untuk setiap baris di mana nilai 'username' tidak sama dengan nilai 'mention'.
- `df[df['username'] != df['mention']]`: Menggunakan series boolean sebagai mask untuk dataframe 'df', sehingga hanya baris-baris yang memenuhi kondisi True (nilai 'username' tidak sama dengan nilai 'mention') yang akan dipertahankan dalam dataframe baru 'df'.
- `df.info()`: Setelah proses penyaringan, `df.info()` digunakan untuk menampilkan informasi mengenai dataframe 'df'.

seperti jumlah baris, jumlah kolom, serta tipe data dari setiap kolom.

- Hasil :

```
<class 'pandas.core.frame.DataFrame'>
Index: 886 entries, 3 to 1291
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype  
---  -
 0   username    886 non-null    object  
 1   mention     886 non-null    object  
dtypes: object(2)
memory usage: 20.8+ KB
```

Langkah ini berguna untuk menghilangkan baris di mana pengguna yang disebutkan dalam teks (kolom 'mention') sama dengan pengguna yang memposting teks (kolom 'username'). Hal ini bisa menjadi langkah penting dalam analisis jaringan sosial untuk menghindari penghitungan interaksi yang tidak relevan antara pengguna dengan diri mereka sendiri.

B. Visualisasi Graph Menggunakan Girvan-Newman Method dan Hasil

1. Setelah data telah melalui proses cleaning, kami akan meimplementasikan metode Girvan-Newman untuk mendeteksi komunitas yang berada pada dataset terkait.

```
def girvan_newman(G):
    def most_valuable_edge(graph):
        centrality = nx.edge_betweenness_centrality(graph, weight='weight')
        return max(centrality, key=centrality.get)

    def without_most_central_edges(graph, most_valuable_edge_func):
        original_num_components = nx.number_connected_components(graph)
        num_new_components = original_num_components

        while num_new_components <= original_num_components:
            edge = most_valuable_edge_func(graph)
            graph.remove_edge(*edge)
            new_components = tuple(nx.connected_components(graph))
            num_new_components = len(new_components)
            return new_components

        if G.number_of_edges() == 0:
            return tuple(nx.connected_components(G))

        g = G.copy().to_undirected()
        g.remove_edges_from(nx.selfloop_edges(g))

        community = []
        modularities = []

        i = 0
        while g.number_of_edges() > 0:
            communities = without_most_central_edges(g, most_valuable_edge)
            modularity = nx.algorithms.community.modularity(G, communities)
            community.append(communities)
            modularities.append(modularity)

            print(f'Count of Community: {len(communities)}, Modularity: {modularity}')

            i += 1
            if len(modularities) > 10 and all(modularities[-i] < modularities[-10] for i in range(2, 10)):
                break

        return community, modularities

community, modularity = girvan_newman(G_main)df = df[df['username'] != df['mention']]
df.info()
```

Kode tersebut mengimplementasikan algoritma Girvan-Newman untuk mendeteksi komunitas dalam sebuah graf berbobot menggunakan NetworkX. Fungsi girvan_newman(G) menghapus tepi dengan nilai sentralitas betweenness tertinggi secara iteratif hingga graf terbagi menjadi beberapa komponen yang tidak terhubung. Selama proses ini, modularitas dihitung dan disimpan setiap kali komunitas baru terbentuk. Perulangan berhenti jika modularitas tidak meningkat secara signifikan atau setelah beberapa iterasi tertentu. Hasil akhirnya adalah daftar komunitas dan nilai modularitas yang terkait.

Kemudian hasil nya seperti ini :

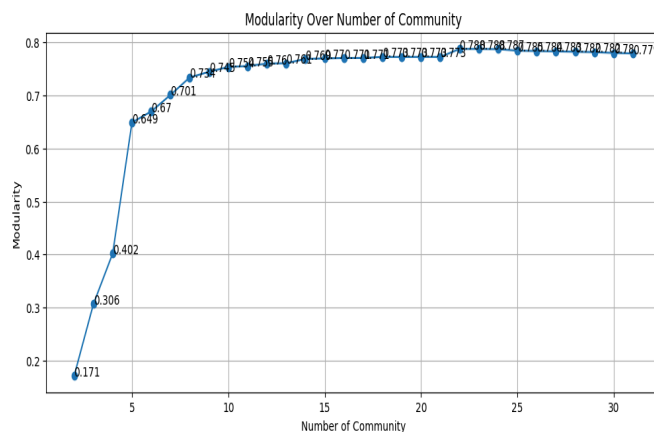
```
Count of Community: 4, Modularity: 0.1709850521963376
Count of Community: 5, Modularity: 0.3063001481207548
Count of Community: 6, Modularity: 0.4018527880338888
Count of Community: 7, Modularity: 0.6489567617222922
Count of Community: 8, Modularity: 0.669837263877223
Count of Community: 9, Modularity: 0.7010439267539916
Count of Community: 10, Modularity: 0.7340209572180836
Count of Community: 11, Modularity: 0.7446215249159566
Count of Community: 12, Modularity: 0.7537153130895078
Count of Community: 13, Modularity: 0.7554463391748906
Count of Community: 14, Modularity: 0.7601555169571709
Count of Community: 15, Modularity: 0.7607938328261559
Count of Community: 16, Modularity: 0.7689817829175265
Count of Community: 17, Modularity: 0.7698138556835685
Count of Community: 18, Modularity: 0.7706154387856065
Count of Community: 19, Modularity: 0.7706449449120619
Count of Community: 20, Modularity: 0.7727044725386482
Count of Community: 21, Modularity: 0.7725825138826327
Count of Community: 22, Modularity: 0.7727182420643274
Count of Community: 23, Modularity: 0.772619888309476
Count of Community: 24, Modularity: 0.7878145598964533
Count of Community: 25, Modularity: 0.7877889879201919
Count of Community: 26, Modularity: 0.7873926222881411
Count of Community: 27, Modularity: 0.7845118408085466
Count of Community: 28, Modularity: 0.783778121797356
Count of Community: 29, Modularity: 0.7831280034777889
Count of Community: 30, Modularity: 0.7824503461068633
Count of Community: 31, Modularity: 0.7817067917201876
Count of Community: 32, Modularity: 0.7802000121958658
Count of Community: 33, Modularity: 0.7792636844496813
```

2. Untuk memperjelas hasil diatas kami membuat struktur program untuk menampilkan graph nya, dengan ini bisa dilihat dengan jelas hasil dari modularity nya.

```
plt.figure(figsize=(12, 5))
plt.plot(range(2, len(modularity) + 2), modularity, marker='o')
for i, txt in enumerate(modularity):
    plt.annotate(round(txt, 3), (i+2, modularity[i]))
plt.xlabel("Number of Community")
plt.ylabel("Modularity")
plt.title("Modularity Over Number of Community")
plt.grid()
plt.show()
```

Kode tersebut menggunakan matplotlib untuk memvisualisasikan perubahan modularitas terhadap jumlah komunitas yang dihasilkan oleh algoritma Girvan-Newman. Fungsi plt.plot

digunakan untuk membuat grafik dengan sumbu x sebagai jumlah komunitas dan sumbu y sebagai nilai modularitas. Setiap titik pada grafik diberi anotasi dengan nilai modularitas yang dibulatkan ke tiga desimal menggunakan `plt.annotate`. Label sumbu dan judul grafik juga ditambahkan untuk memperjelas informasi yang disajikan. Grid pada grafik diaktifkan dengan `plt.grid()` untuk mempermudah pembacaan data. Dari grafik tersebut, dapat disimpulkan bahwa pembentukan komunitas dalam graf cenderung optimal ketika jumlah komunitas berada di sekitar 18, di mana modularitas mencapai nilai tertinggi sekitar 0.788. Ini menunjukkan bahwa pembagian graf ke dalam 18 komunitas memberikan struktur yang paling modular dan



terdefinisi dengan baik. Penambahan lebih lanjut jumlah komunitas tidak memberikan peningkatan yang signifikan dalam modularitas, bahkan dapat menyebabkan sedikit penurunan.

- Setelah itu tahap selanjutnya adalah me-visualisasi akhir dari graph nya dan melihat hasil dari deteksi komunitas dengan menggunakan metode Girvan-Newman.

```
def visualize_gn(G, target_com):
    communities_generator = nx.algorithms.community.girvan_newman(G)
    target_num_communities = target_com

    node_groups = []
    color_map = {}
    centers = {}
    labels = {}

    for i in range(target_num_communities):
        next(communities_generator)

    for com in next(communities_generator):
        node_groups.append(list(com))

    for node in G:
        for i in range(len(node_groups)):
            if node in node_groups[i]:
                color_map[node] = i

    for i, com in enumerate(node_groups):
        center = max(com, key=G.degree)
        centers[center] = node_groups[i]
        labels[center] = center

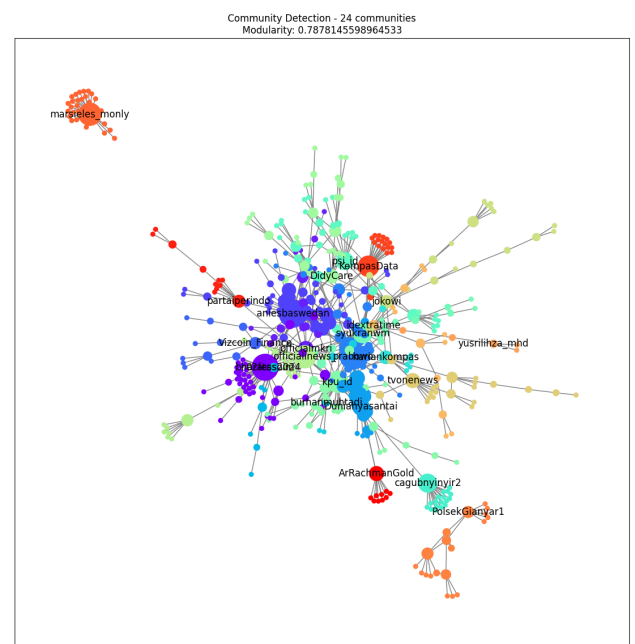
    plt.figure(figsize=(14, 14))
    nx.draw_networkx(
        G,
        labels=labels,
        with_labels=True,
        node_color=color_map,
        cmap=plt.cm.rainbow,
        edge_color='grey',
        linewidth=1,
        font_size=12,
        node_size=1000 for v in dict(G.degree).values():
    )
    pos=nx.fruchterman_reingold_layout(G)

    plt.title('Community Detection - %s communities (modularity: %s)' % (target_com, modularity))
    plt.show()

    return centers

max_index_mod = modularity(G, max_index_mod)
centers = visualize_gn(G, max_index_mod)
```

Kode tersebut digunakan untuk memvisualisasikan hasil deteksi komunitas dalam graf menggunakan algoritma Girvan-Newman dengan NetworkX dan matplotlib. Fungsi `visualize_gn(G, target_com)` menerima dua parameter: graf G dan jumlah target komunitas `target_com`. Fungsi tersebut menjalankan generator komunitas Girvan-Newman hingga mencapai jumlah komunitas yang diinginkan. Kemudian, node dalam graf dikelompokkan berdasarkan komunitas mereka, dan setiap komunitas diberi warna yang berbeda. Node dengan derajat tertinggi dalam setiap komunitas digunakan sebagai label untuk visualisasi. Graf tersebut digambar menggunakan `nx.draw_networkx` dengan tata letak Fruchterman-Reingold.



Hasil visualisasi menunjukkan bahwa algoritma Girvan-Newman berhasil membagi graf menjadi 24 komunitas dengan modularitas yang tinggi, menunjukkan bahwa komunitas-komunitas ini terdefinisi dengan baik. Struktur komunitas yang dihasilkan juga menunjukkan keterpisahan yang signifikan antara komunitas-komunitas, menandakan bahwa algoritma ini efektif dalam mengidentifikasi kelompok node yang saling terhubung erat. Modularitas yang tinggi mengindikasikan bahwa pemisahan ini memiliki kualitas yang baik dalam hal pengelompokan komunitas.

- Kemudian disini kami membagikan 2 kolom data untuk menampilkan dataframe yang menggambarkan komunitas-komunitas yang telah terdeteksi dengan menggunakan metode Girvan-Newman.

	Center	Community
0	bharatasoim	[bharatasoim, msaid_didu, khong_beng, babehald...
1	officialmkri	[f_fathur, vanc1bozz, sectorbee846255, HermanW...
2	aniesbaswedan	[ganjarpranowo", YusSMuh72, Hafiznur_arifin, K...
3	Vizcoin_Finance	[MitraTani54958, sexy_killer404, FootballerVot...
4	prabowo	[gerindra, inyongmanise, Capres_OmonOmon, Bant...
5	Dunianyasantai	[p4p4b0w0_2024, muhsinsukandar6, indobarometer...
6	02fess2024	[iwanpmana87, 02fess2024, Sone__05__, ndaseeee...
7	burhanmuhtadi	[burhanmuhtadi, C012D0N_13LU3, malasnonton, is...
8	hariankompas	[kpu_id], pdi_perjuangan), prabowo)-gibran, gi...
9	cagubnyinyir2	[angelakris_, gundala2221, gajahgemblik, jingg...
10	psi_id	[dibainf, RusliUncle2, miduk17, rianinuraisyah...
11	syukranwm	[syukranwm, AlcoholSwab999, Didi3Ahmadi, chav...
12	kpu_id	[Zaenal58788427, kpu_magetan, Lanoko_, humaspo...
13	DidyCare	[_bangagoengDuA, a_rusman2024, dhemit_is_back...
14	officialnews_	[azerbaijan67, tjabar, ardianzain1, murtadhao...
15	jokowi	[YPbg14669940, vivianrani28, vivacoid, setkabg...
16	tvonenews	[DedenFitri, Sophie99arch, pemilu2024_, theWul...
17	idextratime	[idextratime, alittlejournals, anggahandika20,...
18	yusrilhza_mhd	[tan_mar3m, 1990Husnul, Beritasatu, yusrilhza...
19	PolsekGlanyar1	[sripimpoldajabar, sripimpoldabengkulu, juda...
20	marsieles_monly	[johnreno441996, dhikaa_syahreza, winda_lailas...
21	KompasData	[golkar_id], pimnaspkn), partaigeloraid), dp...
22	partaiperindo	[KunkunTarkun, SINDOnews, okezonenews, aprilia...
23	ArRachmanGold	[ArRachmanGold, axelthethinker, kataralastair0...

Gambar disini menunjukkan hasil deteksi komunitas menggunakan metode Girvan-Newman, yang membagi jaringan menjadi beberapa komunitas berdasarkan penghapusan bertahap sisi dengan betweenness tertinggi. Tabel tersebut menampilkan dua kolom utama: "Center" yang menunjukkan node pusat setiap komunitas, dan "Community" yang berisi daftar node yang tergabung dalam komunitas tersebut. Contohnya, komunitas dengan pusat "bharatasoim" mencakup anggota seperti "msaid_didu" dan "khong_beng", sedangkan komunitas yang dipimpin oleh "officialmkri" mencakup anggota seperti "f_fathur" dan "vanc1bozz".

IV. KESIMPULAN

Dalam penelitian ini, metode Girvan-Newman berhasil diterapkan untuk mendeteksi komunitas dalam jaringan sosial yang dibentuk oleh interaksi pemilih dan kandidat pada Pemilu 2024. Melalui algoritma ini, kami mampu mengidentifikasi 24 komunitas dengan modularitas yang tinggi, yang menunjukkan bahwa komunitas-komunitas tersebut terdefinisi dengan baik. Struktur komunitas yang dihasilkan memperlihatkan keterpisahan signifikan antar

komunitas, menandakan efektivitas algoritma dalam mengidentifikasi kelompok node yang saling terhubung erat. Dengan modularitas tinggi, pembagian ini menunjukkan kualitas yang baik dalam pengelompokan komunitas, memberikan wawasan berharga untuk memahami dinamika politik dan strategi kampanye yang lebih efektif. Penelitian ini memberikan kontribusi penting dalam analisis jaringan sosial dan studi politik, serta dapat menjadi acuan untuk penelitian lebih lanjut dan perencanaan strategis oleh partai politik.

V. LAMPIRAN

- LINK GOOGLE COLLAB :

 [PengorbananAirMata_dan_Darah.ipynb](#)

VI. DAFTAR PUSTAKA

- Vivifriskilangela. (2023). ANALISIS PERAN MEDIA SOSIAL DALAM PENGARUH POLITIK MENJELANG PEMILU. *Wacana: Jurnal Ilmu Sosial Dan Ilmu Politik Interdisiplin*, 10(1), 555–564.
<https://doi.org/10.37304/wacana.v10i1.9604>
- Rojabiaturohmah, E., & Pujiarti, L. S. P. (2020). Keterlibatan Media Sosial Sebagai Media Informasi Menjelang Pemilu 2019. *Jurnal Pendidikan Politik, Hukum Dan Kewarganegaraan*, 10(1). Retrieved June 14, 2024, from Wacana : Jurnal Ilmu Sosial dan Ilmu Politik Interdisiplin.
<https://doi.org/10.35194/jpphk.v10i1.932>
- Javed, M. A., Younis, M. S., Latif, S., Qadir, J., & Baig, A. (2018, April 15). Community detection in networks: A multidisciplinary review. *Journal of Network and Computer Applications*. Academic Press.
<https://doi.org/10.1016/j.jnca.2018.02.011>
- Shahmoradi, M. R., Ebrahimi, M., Heshmati, Z., & Salehi, M. (2019). Multilayer overlapping community detection using multi-objective optimization. *Future Generation Computer Systems*, 101, 221–235.
<https://doi.org/10.1016/j.future.2019.05.061>
- Rich, M. J., & Pather, S. (2021). A response to the persistent digital divide: Critical components of a community network ecosystem. *Information Development*, 37(4), 558–578.
<https://doi.org/10.1177/0266666920924696>