

# Penerapan Teori Graf dalam Mendeteksi Pola Penipuan pada Jaringan Komunikasi

Stefan Matthew Susanto - 13523020

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

[stefanmattew246@gmail.com](mailto:stefanmattew246@gmail.com), [13523020@std.stei.itb.ac.id](mailto:13523020@std.stei.itb.ac.id)

**Abstrak**— Makalah ini membahas penerapan teori graf dalam mendeteksi pola penipuan pada jaringan komunikasi. Dengan memanfaatkan konsep graf seperti graf sederhana, graf berarah, dan keterhubungan antar simpul. Penelitian ini bertujuan untuk mengidentifikasi nomor-nomor telepon yang terlibat dalam aktivitas penipuan. Dataset yang digunakan terdiri dari pasangan nomor telepon dan tag yang merepresentasikan hubungan antar nomor, yang kemudian divisualisasikan menggunakan teori graf untuk mempermudah analisis. Hasil penelitian menunjukkan bahwa pendekatan berbasis teori graf efektif dalam mengungkap pola keterhubungan yang mencurigakan.

**Keywords**—Teori graf, pola penipuan, jaringan komunikasi, analisis graf

## I. PENDAHULUAN

Di era digital ini, salah satu media komunikasi dengan telepon menjadi bagian penting dari kehidupan sehari-hari. Sayangnya banyak pihak-pihak yang menyalahgunakan untuk melakukan penipuan. Metode tradisional untuk mendeteksi penipuan seringkali kurang efisien karena keterbatasan dalam mengolah data besar yang kompleks. Oleh karena itu, diperlukan pendekatan baru untuk mengidentifikasi pola-pola penipuan secara efektif.

Makalah ini bertujuan untuk memanfaatkan teori graf dalam mendeteksi pola penipuan pada jaringan komunikasi. Teori graf memungkinkan representasi visual dari hubungan antar nomor telepon yang dapat mempermudah analisis dan pengambilan keputusan. Dengan menggunakan dataset yang berisi pasangan nomor telepon dan tag, makalah ini menganalisis bagaimana konsep seperti graf sederhana, graf berarah, dan derajat simpul dapat digunakan untuk mengidentifikasi nomor-nomor yang terlibat dalam aktivitas penipuan.

## II. TEORI DASAR

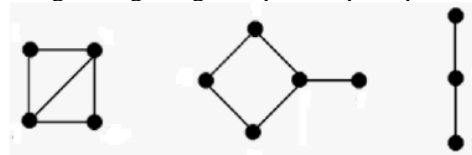
### 2.1. Graf

Graf umumnya digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut. Secara umum, Graf  $G$  didefinisikan sebagai  $G = (V, E)$  dengan  $V$  merupakan himpunan tidak kosong dari simpul-simpul didefinisikan sebagai  $V = \{v_1, v_2, v_3, \dots, v_n\}$ , dan  $E$  merupakan himpunan sisi yang menghubungkan sepasang simpul didefinisikan sebagai  $E = \{e_1, e_2, e_3, \dots, e_n\}$ . Graf harus memiliki minimal 1 simpul, sedangkan himpunan  $E$  boleh tidak memiliki sisi satu pun.

Jenis dapat dibagi menjadi 2 berdasarkan ada atau tidaknya sisi di dalam graf:

#### 1. Graf Sederhana (*simple graf*)

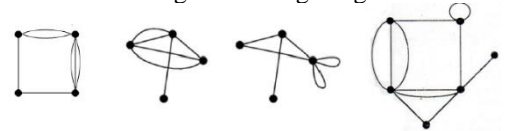
Graf sederhana adalah graf yang tidak mengandung sisi ganda pada tiap simpul.



Gambar 1 Graf Sederhana. Sumber PPT Matdis Rinaldi Munir

#### 2. Graf Tak-sederhana

Graf tak-sederhana adalah graf yang mengandung minimal satu sisi ganda atau gelang

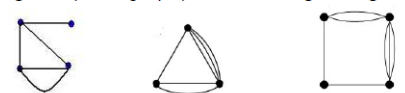


Gambar 2 Graf tak-sederhana. Sumber PPT Matdis Rinaldi Munir

Graf tak-sederhana dapat dibedakan kembali menjadi 2 yaitu

#### 1. Graf ganda (*multi-graph*)

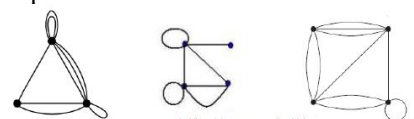
Graf ganda adalah graf yang mengandung sisi ganda. Sisi ganda ditunjukkan dengan terdapatnya 2 atau lebih sisi yang menghubungkan pasangan simpul yang sama



Gambar 3 Graf Ganda. Sumber PPT Matdis Rinaldi Munir

#### 2. Graf semu (*pseudo-graph*)

Graf semu adalah graf yang mengandung sisi gelang. Sisi gelang adalah sisi yang menghubungkan suatu simpul dengan simpul itu sendiri.



Gambar 4 Graf Semu. Sumber PPT Matdis Rinaldi Munir

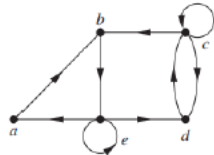
Graf dapat dibedakan menjadi 2 jenis berdasarkan orientasi arah pada sisi:

1. Graf tak-berarah (*undirect graph*)  
Graf tak-berarah adalah graf yang tidak mempunyai orientasi arah.



Gambar 5 Graf tak-berarah. Sumber PPT Matdis Rinaldi Munir

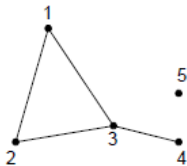
2. Graf berarah (*directed graph*)  
Graf berarah adalah graf yang memiliki orientasi arah pada tiap sisi.



Gambar 6 Graf Berarah. Sumber PPT Matdis Rinaldi Munir

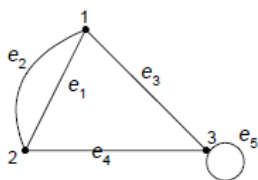
**Terminologi:**

1. Bertetangga (*Adjacent*)  
Jika dua simpul terhubung langsung maka kedua simpul tersebut dikatakan bertetangga. Pada Gambar 7, Simpul 1 bertetangga dengan simpul 2 dan 3. Namun simpul 1 tidak bertetangga dengan simpul 4 dan 5.



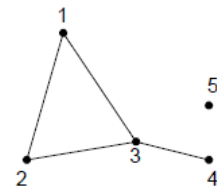
Gambar 7 Graf Adjacent. Sumber PPT Matdis Rinaldi Munir

2. Bersisian (*incidency*)  
Suatu sisi pada graf akan bersisian dengan tiap simpul yang dihubungkannya. Pada Gambar 8,  $e_1$  akan bersisian dengan simpul 1 dan simpul 2. Dan  $e_3$  akan bersisian dengan simpul 1 dan simpul 3.



Gambar 8 Graf Incidency. Sumber PPT Matdis Rinaldi Munir

3. Simpul terpecil (*isolated vertex*)  
Simpul terpecil adalah simpul yang tidak memiliki sisi yang bersisian dengan simpul tersebut. Pada Gambar 9, Simpul 5 merupakan simpul terpecil karena tidak memiliki sisi yang menghubungkan simpul itu dengan simpul yang lain.



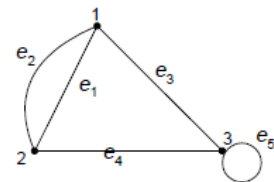
Gambar 9 Simpul Terpecil. Sumber PPT Matdis Rinaldi Munir

4. Graf Kosong (*null graph*)  
Graf kosong adalah graf yang tidak memiliki sisi satu pun.



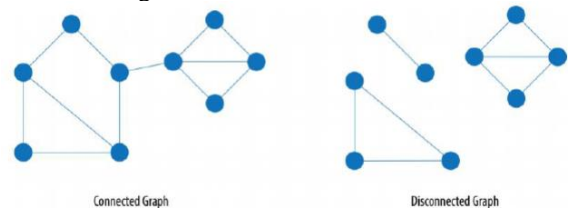
Gambar 10 Graf Kosong. Sumber PPT Matdis Rinaldi Munir

5. Derajat (*degree*)  
Derajat pada suatu simpul ialah jumlah sisi yang bersisian dengan simpul itu. Pada Gambar 11, jumlah derajat simpul 1 adalah 3 karena simpul 1 memiliki 3 sisi yakni  $e_1, e_2, e_3$ .



Gambar 11 Graf Derajat. Sumber PPT Matdis Rinaldi Munir

6. Lintasan (*path*)  
Lintasan adalah barisan berselang-seling simpul dan sisi yang ditempuh untuk mencapai simpul tujuan dari simpul awal. Sedangkan Panjang lintasan adalah jumlah sisi yang diperlukan dalam lintasan tersebut.
7. Siklus (*cycle*)  
Siklus adalah lintasan yang memiliki simpul awal dan akhir yang sama.
8. Keterhubungan (*connected*)  
Keterhubungan adalah graf yang memiliki lintasan antar simpul dengan simpul lain. Suatu graf disebut graf terhubung jika graf tersebut saling menghubungkan pada setiap simpul. Jika terdapat simpul yang tidak terhubung maka graf tersebut graf tak-terhubung.



Gambar 12 Graf Connected. Sumber PPT Matdis Rinaldi Munir

Selain beberapa jenis sebelumnya, terdapat beberapa jenis graf khusus yakni:

- a. Graf Lengkap (*complete graph*)



```
# membuat dataframe dari dataset
df = pd.DataFrame(data)
```

Gambar 19 Membuat dataframe dari dataset

Program akan membuat dataframe dari dataset dengan menggunakan pandas.

```
# Mencari tag yang berisi "Penipu"
penipu_data = df[df['tags_1'].apply(lambda tags: 'Penipu' in tags) | df['tags_2'].apply(lambda tags: 'Penipu' in tags)]
```

Gambar 20 Mencari tag yang berisi penipu

Program akan membuat data baru bernama penipu\_data. Data ini terdiri dari nomor yang memiliki tags 1 ataupun tags 2 sebagai 'penipu'.

### 3.2. Pembentukan Graf

```
# membuat graf jaringan penipu
G = nx.Graph()

for idx, row in penipu_data.iterrows():
    G.add_edge(row['number_1'], row['number_2'], tags_1=row['tags_1'], tags_2=row['tags_2'])
```

Gambar 21 Membuat graf jaringan penipu

Program akan membuat graf baru yakni graf jaringan penipu. Program akan membuat simpul baru dari nomor 1 dan 2 dan tags 1 dan 2.

```
# Menambah simpul graf
for node in G.nodes():
    tags = set()
    for idx, row in penipu_data.iterrows():
        if row['number_1'] == node:
            tags.update(row['tags_1'])
        if row['number_2'] == node:
            tags.update(row['tags_2'])
    G.nodes[node]['tags'] = ', '.join(tags)
```

Gambar 22 Menambah simpul graf

Program akan memproses data dari penipu\_data dan menghubungkan ke simpul-simpul dalam graf G. setelah itu akan ditambah informasi mengenai tiap simpul tersebut.

```
# menentukan warna simpul berdasarkan tag
node_colors = []
for node, attr in G.nodes(data=True):
    if 'Penipu' in attr['tags']:
        # warna penipu diberi warna merah
        node_colors.append('red')
    else:
        # warna bukan penipu tidak diberi merah
        node_colors.append('skyblue')
```

Gambar 23 Menentukan warna simpul berdasarkan tag

Program akan menentukan warna simpul berdasarkan tag nya. Jika simpul memiliki tag penipu maka warna dari simpul tersebut akan berubah menjadi merah. Sedangkan warna jika tidak memiliki tag penipu maka akan berubah menjadi biru.

### 3.3. Visualisasi Graf

```
# Menampilkan graf
plt.figure(figsize=(10, 7))
pos = nx.spring_layout(G, seed=42)
nx.draw(
    G,
    pos,
    with_labels=True,
    node_size=700,
    node_color=node_colors,
    font_size=10,
    font_color='black',
)

plt.show()
```

Gambar 24 Menampilkan graf

Graf yang dibuat akan ditampilkan dengan matplotlib dan terdapat beberapa modifikasi supaya tampilan dapat cocok dan mudah untuk dianalisis.

### 3.4. Deteksi nomor penipu

```
# Daftar Penipu
fraud_list = []
for node, attr in G.nodes(data=True):
    if 'Penipu' in attr['tags']:
        fraud_list.append(node)

print("Daftar Penipu:", fraud_list)
```

Gambar 25 Deteksi nomor penipu

Program akan memasukkan list nomor penipu ke dalam fraud\_list. Program akan mengecek apakah terdapat tag 'penipu' atau tidak dari graf tersebut.

```
# Fungsi mencari tag berdasar nomor
def search_tags(number):
    tags = set()
    for idx, row in penipu_data.iterrows():
        if row['number_1'] == number:
            tags.update(row['tags_1'])
        if row['number_2'] == number:
            tags.update(row['tags_2'])
    return list(tags)
```

Gambar 26 Mencari tag berdasar nomor

Program akan mencari tag yang dimiliki nomor 1 atau nomor 2 dari list penipu.

```
# mengetes nomor dengan fungsi search_tags
input_number = '0'
while input_number != 'q':
    input_number = (input("Masukkan nomor telepon(ketik 'q' untuk quit): "))

    result_tags = search_tags(input_number)

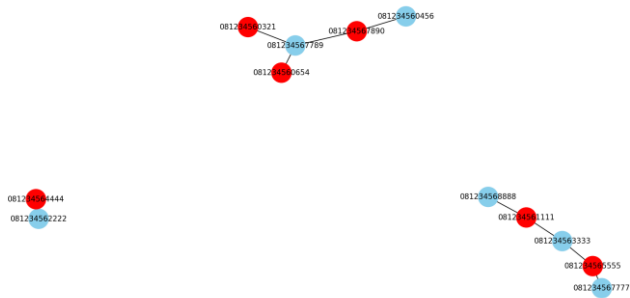
    if result_tags:
        print(f"Tag untuk nomor {input_number}: {result_tags}")
    else:
        print(f"Tidak ditemukan tag untuk nomor {input_number}.")
```

Gambar 27 Mengetes nomor dengan fungsi search\_tags

Program bisa menampilkan hasil tag yang dimiliki oleh tiap nomor dari dataset. Pengguna bisa memasukkan nomor yang akan dicek, lalu program akan mencari nomor dari fraud\_list dan menampilkan semua tag yang dimiliki. Jika tidak terdapat nomor yang dicari maka akan memberikan output “tidak ditemukan tag”. Proses ini sama seperti cara kerja aplikasi GetContact dimana get contact bisa mencari nomor dan akan menampilkan tag yang diinginkan.

#### IV. HASIL DAN PEMBAHASAN

Setelah proses implementasi selesai, program akan memberikan output dan hasil yang dapat digunakan untuk analisis.

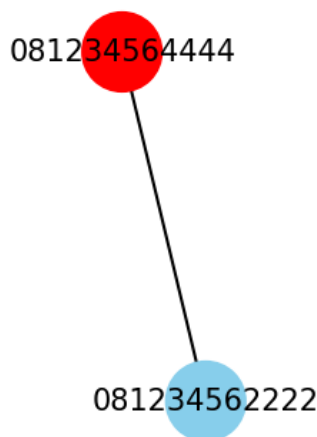


Gambar 28 Hasil graf dataset 1

Dengan dataset 14 nomor terdapat 3 kelompok graf terpisah. Graf yang dibentuk terdapat 2 warna simpul yaitu merah dan biru. Simpul yang berwarna merah menandakan simpul penipu. Sedangkan simpul yang berwarna biru adalah simpul orang

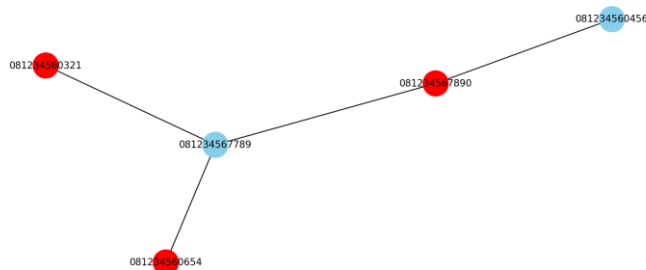
bukan penipu. Hubungan graf tersebut ditunjukkan dengan sisi. Sisi graf tersebut menunjukkan keterhubungan antara number-1 dan number-2. Jika number\_1 dan number\_2 saling berhubungan maka mereka berdua akan saling menyimpan nomor satu sama lain dengan tag masing-masing.

alam graf tersebut hanya terdapat 12 simpul karena 2 simpul sisanya merupakan orang yang tidak berinteraksi dengan penipu, maka tidak terdapat di dalam graf.



Gambar 29 Komponen pertama graf dataset 1

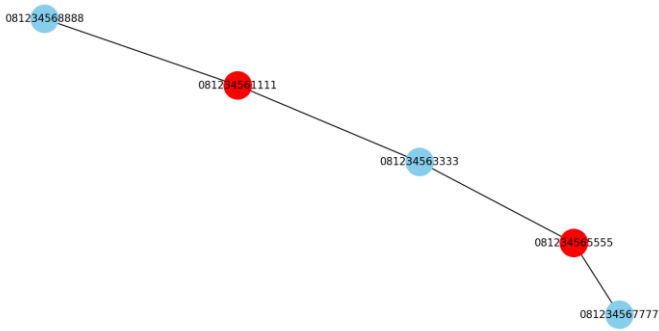
Pada graf sebelah kiri bawah, terdapat 2 simpul yang saling berhubungan. Nomor ‘08123456444’ berhubungan dengan nomor ‘08123456222’. Simpul dengan nomor ‘08123456444’ memiliki warna merah, menandakan bahwa nomor tersebut merupakan nomor penipu yang telah mengontak nomor ‘08123456222’. Orang bernomor ‘08123456222’ menamai nomor ‘08123456444’ di kontakannya sebagai “penipu”.



Gambar 30 Komponen kedua graf dataset 1

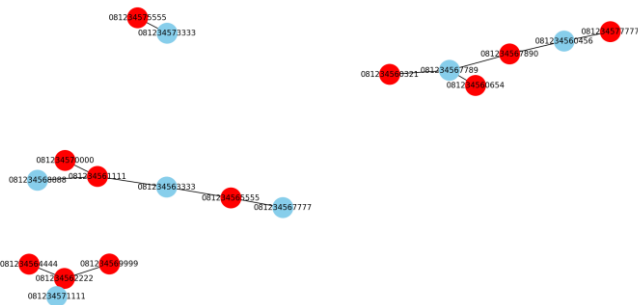
Pada Gambar 30, nomor ‘081234567789’ berhubungan dengan 3 penipu sekaligus yakni ‘081234560654’, ‘081234560321’, ‘081234567890’ dan telah menamai mereka semua dengan nama “penipu” di kontakannya. Sedangkan nomor ‘081234560456’ berhubungan dengan penipu ‘0812334567890’ namun tidak berhubungan dengan penipu lainnya.





Gambar 31 Komponen ketiga graf dataset 2

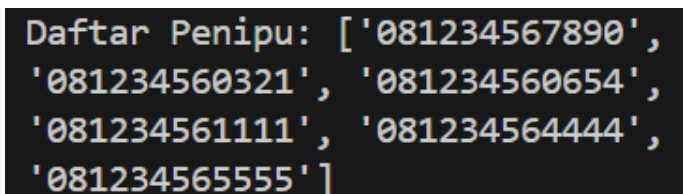
Pada Gambar 31, nomor '08123456888' menyimpan nomor '081234561111' sebagai penipu. Lalu nomor '081234563333' telah ditipu oleh 2 penipu yakni '081234561111' dan '081234565555'. Nomor '081234565555' juga menipu nomor '081234567777'.



Gambar 32 Hasil graf dataset 2

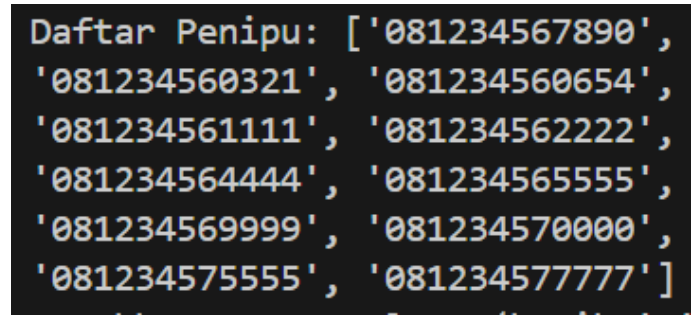
Pada Gambar 32, graf tersebut menggunakan dataset yang lebih banyak dari dataset sebelumnya. Dari graf tersebut dapat dilihat terdapat 4 kelompok graf yang terpisah. Dalam graf tersebut menunjukkan terdapat total 11 penipu.

#### 4.2. Hasil Daftar Penipu



Gambar 33 Hasil daftar penipu dataset 1

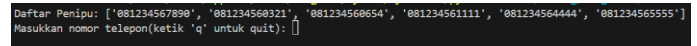
Program tersebut akan menampilkan daftar nomor semua penipu dari dataset. Dataset yang digunakan pada Gambar 33 menggunakan dataset yang kecil.



Gambar 34 Hasil daftar penipu dataset 2

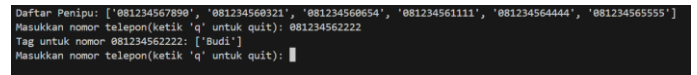
Sedangkan pada gambar X.X., terdapat 11 penipu dalam dataset tersebut.

#### 4.3. Cek Nomor



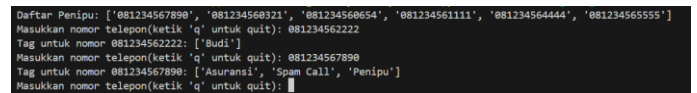
Gambar 35 Program meminta input pengguna

Setelah berhasil menampilkan daftar penipu, pengguna/user bisa melakukan pengecekan nomor, apakah nomor tersebut merupakan nomor penipu atau bukan.



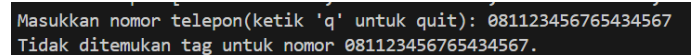
Gambar 36 Pengguna memasukkan nomor bukan penipu

Pada Gambar 36, pengguna memasukkan nomor bukan penipu sehingga terdapat tag nama orang tersebut bukan tag 'penipu'.



Gambar 37 Pengguna memasukkan nomor penipu

Pada Gambar 37, pengguna memasukkan nomor penipu, maka pada output yang diberikan terdapat salah satu tag 'penipu' dari semua tag yang dimiliki nomor itu.



Gambar 38 Pengguna memasukkan nomor yang tidak ada di dataset

Pada Gambar 38, pengguna memasukkan nomor yang tidak ada di database maka program akan menampilkan output bahwa tidak ada tag ditemukan.

## V. KESIMPULAN

Penerapan teori graf menjadi salah satu metode yang cukup efektif untuk mendeteksi pola penipuan pada jaringan komunikasi. Analisis graf memungkinkan identifikasi nomor telepon yang mencurigakan berdasarkan hubungan antar simpul dan pola keterhubungannya. Dengan pendekatan ini, pola penipuan dapat diungkap dengan lebih cepat dan akurat dibandingkan metode tradisional. Penelitian ini juga memberikan landasan bagi pengembangan sistem deteksi

otomatis berbasis teori graf yang dapat diintegrasikan ke dalam sistem komunikasi modern.

## VI. LAMPIRAN

Lampiran Github dalam penelitian ini  
<https://github.com/StefanMattew/Graph-in-fraud-detection.git>

## VII. UCAPAN TERIMA KASIH

Ucapan terimakasih kepada keluarga tercinta dan sahabat yang telah memberikan dukungan kepada penulis dalam menyelesaikan tugas berikut.

### REFERENCES

- [1] R. Munir, Graf Bagian 1, 2024. [Daring]. Tersedia: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf> [Diakses: 6-Jan-2025].
- [2] R. Munir, Graf Bagian 2, 2024. [Daring]. Tersedia: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian2-2024.pdf> [Diakses: 6-Jan-2025].
- [3] R. Munir, Graf Bagian 3, 2024. [Daring]. Tersedia: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian3-2024.pdf> [Diakses: 6-Jan-2025].
- [4] Rogers, E. M. (1986). Communication Technology: The New Media in Society. New York: Free Press

### PERNYATAAN

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

Bandung, 8 Januari 2025



Stefan Mattew Susanto - 13523020