Analyzing Indonesian YouTube Trending Content through Graph Theory Approach

Muhammad Raihaan Perdana - 13523124[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]perdanaraihan96@gmail.com, 13523124@std.stei.itb.ac.id

***Abstract*—This paper examines the patterns and relationships within Indonesia's YouTube trending content through a graph-theoretical framework, where videos are modeled as vertices connected by weighted edges representing shared channel ownership (weight=1.0), content categories (weight=0.7), and temporal relationships (weight=0.5). Through computational analysis of two weeks of daily trending data, the study constructs a complex network with 50 nodes and 1,275 edges, achieving a graph density of 1.0408 and perfect clustering coefficient of 1.0000. Using NetworkX for network analysis, the research implements degree centrality, betweenness centrality, and clustering algorithms to identify influence patterns. The resulting network reveals distinct content communities, with centrality measurements highlighting influential videos achieving centrality scores of 1.0408. Category analysis through graph relationships exposes content preferences in the Indonesian market, showing dominance of People & Blogs (32%) and Gaming (28%) categories. The study demonstrates how graph theory can effectively quantify and visualize complex relationships in social media content, providing insights for content creators and platform operators while establishing a mathematical foundation for analyzing digital content ecosystems.**

***Keywords*— Content communities, Graph theory, Network analysis, YouTube trending.**

I. Introduction

Social media platforms have revolutionized how Indonesians create and consume content, with YouTube emerging as a central player in the country's digital landscape. Understanding the mechanisms behind content popularity and trending patterns has become crucial for content creators, marketers, and platform operators. The platform's trending page serves as a powerful force in shaping online conversations and content creation strategies, making it an ideal subject for data-driven analysis through graph theory – a mathematical framework well-suited for studying interconnected systems.

This research analyzes two weeks of YouTube Indonesia's top 50 trending videos through graph theory principles. The study collects key metrics including video titles, channel information, view counts, likes, and content categories through YouTube's data API and specialized scraping tools. By representing each video as a vertex in a graph and establishing edges based on shared characteristics, the analysis aims to uncover meaningful patterns in how content achieves and maintains trending status.

The methodology leverages Python's ecosystem of data analysis tools. NetworkX provides the foundation for graph construction and analysis, while specialized YouTube scraping libraries enable efficient data collection. Gephi serves as the primary tool for visualizing the resulting networks, helping transform complex data relationships into interpretable patterns. This technical framework allows for processing large amounts of trending video data while maintaining analytical rigor.

Through graph theoretical concepts, the study examines how different types of content interact within Indonesia's YouTube trending ecosystem. The analysis focuses on identifying content clusters, measuring channel influences, and understanding temporal patterns of trending videos. These insights benefit content creators optimizing their strategies, platform operators understanding content dynamics, and researchers studying digital content patterns. With Indonesia's growing digital presence, understanding these patterns becomes increasingly valuable for stakeholders across the digital content ecosystem.

The following sections detail the methodology for data collection and graph construction, present findings about network structure and content patterns, and discuss implications for understanding Indonesia's YouTube trending ecosystem.

II. Theoretical Foundation

A. Graph Definition and Fundamentals

A graph is a mathematical structure that represents relationships between objects. While this might sound complex, we encounter graphs daily - from social media connections to road networks. In its simplest form, a graph consists of vertices (also called nodes) connected by edges. Formally, a graph G is defined as an ordered pair $G = (V, E)$ where V is a non-empty set of vertices and E is a set of edges connecting pairs of vertices [1].

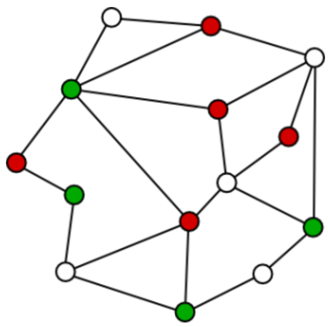


Fig 2.1 Graph Illustration
(Source: open.ac.uk)

Think of YouTube trending videos as vertices, with edges representing meaningful relationships between them - similar content, shared viewers, or same channel origin. Graphs can take different forms depending on their properties:

- Directed graphs: edges have direction (like one video linking to another)
- Undirected graphs: edges have no direction (like videos sharing viewers)
- Simple graphs: no self-loops or multiple edges between vertices
- Multigraphs: allowing multiple edges between vertices.

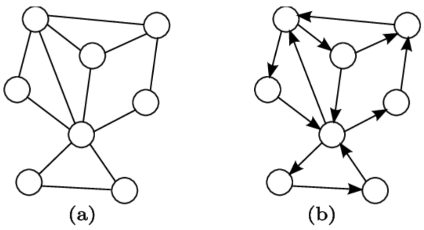


Fig 2.2. a) undirected graph b) directed graph
(Source: researchgate.net)

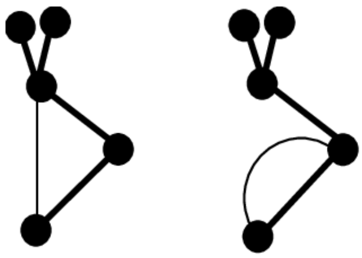


Fig 2.3 Simple graph (left) Multigraphs (right)
(Source: researchgate.net)

B. Weighted-Graph

A weighted graph is a graph in which each edge has an associated numerical value called a weight. In the context of social media analysis, these weights can represent the strength of relationships between content pieces [2].

Formally, a weighted graph G can be defined as an ordered triple $G = (V, E, w)$ where:

- V is a set of vertices or nodes
- E is a set of edges connecting pairs of vertices
- w is a weight function that assigns a real number to each edge.

For any edge e ∈ E that connects vertices u and v, we denote its weight as w(e) or w(u,v). The weight function $w: E \to R$ maps each edge to a real number that represents some meaningful quantity in the context of the problem being modeled.

In a weighted graph, the weight of a path $P = (v_1, v_2, \dots, v_k)$ is computed as the sum of the weights of its constituent edges: $w(P) = \sum_{i=1}^{k-1} w(v_i, v_{i+1})$

Weighted graphs find extensive applications in various domains:

- Transportation networks, where weights represent distances or travel times.
- Communication networks, where weights indicate bandwidth or latency.
- Social networks, where weights measure relationship strengths.
- Content analysis systems, where weights represent similarity measures [3].

For YouTube trending analysis, weights can represent various relationships between videos such as:

- Similarity of content categories
- Viewer overlap percentages
- Temporal proximity in trending charts
- Engagement pattern correlations

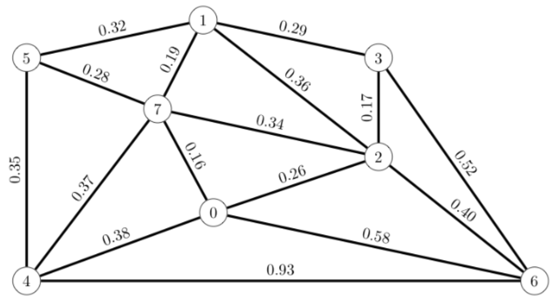


Fig 2.4 Weighted-Graph
(Source: mathcenter.oxford.emory.edu)

These weights help quantify and analyze the complex relationships between trending videos, enabling deeper understanding of content popularity patterns.

C. Graph Centrality

In graph theory, centrality measures are indicators that help identify the most important or influential vertices within a graph. These measures are particularly useful in analyzing social networks and content relationships [4].

1. Degree Centrality

Degree centrality is the simplest form of node centrality, measuring the total number of edges connected to a node. For a vertex v in a graph with n vertices, the degree centrality is defined as:

$$CD(v) = \frac{\deg(v)}{n-1}$$

where $\deg(v)$ is the number of edges incident to v, and $(n-1)$ is the normalization factor. In the context of trending videos, a high degree centrality indicates that a video has many connections to other trending content.

2. Betweenness Centrality

Betweenness centrality measures the extent to which a vertex lies on paths between other vertices. For a vertex v, it is defined as:

$$CB(v) = \sum_{s \neq v \neq t} \frac{\sigma st(v)}{\sigma st}$$

where σst is total number of shortest paths from node s to node t and $\sigma st(v)$ is number of those paths passing through v.

In trending analysis, high betweenness centrality might indicate videos that bridge different content categories or viewer communities.

3. Closeness Centrality

Closeness centrality measures how close a vertex is to all other vertices in the graph. For a vertex v, it is calculated as:

$$CC(v) = \frac{1}{\sum_{t \in V} d(v,t)}$$

where $d(v,t)$ is the shortest path distance between vertices v and t. In the context of YouTube trends, closeness centrality can identify videos that are central to the overall trending ecosystem [5].

These centrality measures provide different perspectives on the importance of nodes within the network, helping identify key patterns and influential content in the trending video landscape.

D. YouTube Platform

YouTube's trending section showcases videos that have gained significant momentum in terms of viewer engagement and popularity within a specific region. Videos typically reach trending status through a combination of factors including rapid view accumulation, high engagement rates, and content novelty. The platform's trending algorithm considers multiple variables such as view velocity (rate of view count increase), user engagement patterns, and regional relevance, though the exact weighting of these factors remains proprietary to YouTube. Trending videos often share characteristics such as timely content, broad audience appeal, and strong initial engagement metrics.

The platform measures video performance through several key metrics that indicate content success and viewer interaction. View count serves as the primary metric, representing the total number of times a video has been watched. The like ratio (likes compared to total views) and engagement rate (combination of likes, comments, and shares relative to views) provide deeper insights into audience response. YouTube also categorizes videos into predefined content categories such as Entertainment, Music, Gaming, and News & Politics, allowing for systematic content organization and analysis. These metrics collectively enable quantitative analysis of video performance and trending patterns.

E. Pattern Analysis

Clustering in social media analysis refers to the process of grouping similar content based on shared characteristics. In the context of video content analysis, clustering helps identify natural groupings of videos that share common attributes such as viewer demographics, engagement patterns, or thematic elements. Graph-based clustering specifically leverages network structure to detect communities of interconnected content, where edges represent meaningful relationships between videos such as shared audiences or similar engagement patterns [6].

Time series analysis in social media examines how content performance evolves over time, which is particularly crucial for understanding trending patterns. Viral content typically exhibits characteristic temporal signatures: rapid initial growth, peak visibility period, and gradual decay in engagement. The duration of trending status varies significantly across content types, with some videos maintaining trending status for several days while others peak briefly. Peak analysis helps identify factors contributing to maximum engagement, including timing of publication, external events, and content characteristics that resonate with viewers. Understanding these temporal patterns provides valuable insights for content strategy and trending prediction [7].

III. Methodology

A. Data Collection and Processing

This research uses a systematic approach to collect and process trending videos from YouTube Indonesia, leveraging computational methods and graph theory principles.

1. Data Acquisition

   The data collection process utilizes the YouTube Data API v3 with the following key characteristics:

   - Collection Scope: Top 50 trending videos in Indonesia
   - Time Frame: Consecutive 14-day period
   - Data Metrics Collected:
     - Video identification: ID, title
     - Channel information: Channel ID, title
     - Performance metrics: View count, like count, comment count
     - Categorical data: Category ID and name
     - Temporal information: Publication time, collection date

   The data collection follows a structured method mathematically represented as:

   $$D = \{v_i | i = 1,2,\dots,n\}$$

   Where:

   - D represents the dataset of trending videos
   - v_i represents an individual video
   - n is the total number of videos (maximum 50 per day)

2. Data Processing Techniques

   Data processing involves several critical steps:

   - Normalization of textual data
   - Category mapping using YouTube's predefined category IDs
   - Handling of missing values and potential duplicates
   - Conversion to structured pandas DataFrame

```python
def __init__(self, api_key):
    self.youtube = build('youtube', 'v3', developerKey=api_key)

def get_category_names(self):
    try:
        request = self.youtube.videoCategories().list(
            part='snippet',
            regionCode='ID'
        )
        response = request.execute()
        return {item['id']: item['snippet']['title']
                for item in response['items']}
    except Exception as e:
        print(f"Error fetching categories: {str(e)}")
        return {}

def collect_trending_videos(self, days=14):
    # Collect trending videos for specified number of days
    all_data = []
    end_date = datetime.now()
    start_date = end_date - timedelta(days=days)
    categories = self.get_category_names()

    current_date = start_date
    while current_date <= end_date:
        try:
            videos = self._get_daily_trending(categories)
            if videos:
                df = pd.DataFrame(videos)
                df['collection_date'] = current_date.date()
                all_data.append(df)
        except Exception as e:
            print(f"Error collecting data for {current_date.date()}: {str(e)}")
        current_date += timedelta(days=1)
```

Fig 3.1 data_collector.py (1)
(Source: Screenshot by the Author)

```python
def _get_daily_trending(self, categories, max_results=50):
    request = self.youtube.videos().list(
        part='snippet,statistics,contentDetails',
        chart='mostPopular',
        regionCode='ID',
        maxResults=max_results
    )
    response = request.execute()

    return [{
        'video_id': item['id'],
        'title': item['snippet']['title'],
        'channel_id': item['snippet']['channelId'],
        'channel_title': item['snippet']['channelTitle'],
        'category_id': item['snippet']['categoryId'],
        'category_name': categories.get(item['snippet']['categoryId'], 'Unknown'),
        'view_count': int(item['statistics']['viewCount']),
        'like_count': int(item['statistics'].get('likeCount', 0)),
        'comment_count': int(item['statistics'].get('commentCount', 0)),
        'publish_time': item['snippet']['publishedAt']
    } for item in response['items']]
```

Fig 3.2 data_collector.py (2)
(Source: Screenshot by the Author)

B. Graph Construction Methodology

The research models the trending video ecosystem as a complex network, where videos are represented as vertices and relationships as edges.

1. Graph Theoretical Framework
   Formally the graph G is defined as:
   $$G = (V, E, W)$$
   Where:
   - V is the set of vertices (videos)
   - E is the set of edges representing relationships
   - W is the weight function mapping edges to relationships strengths

   Vertex Representation:
   - $title(v_i)$: Video title
   - $channel(v_i)$: Channel name
   - $category(v_i)$: Content category
   - $views(v_i)$: View count

   Edge Creation Strategies
   Three primary relationship types define edge creation:
   - Channel Relationship (weight = 1.0):
     Connects videos from the same channel
     $$w_{channel}(e) = 1.0$$
   - Category Relationship (weight = 0.7):
     Connects videos within the same content category
     $$w_{category}(e) = 0.7$$
   - Temporal Relationship (weight = 0.5):
     Connects videos trending on the same date
     $$w_{temporal}(e) = 0.5$$
2. Network Construction Algorithm
   The graph construction process follows these steps:
   - Add video nodes with comprehensive attributes
   - Establish channel-based connections
   - Create category-based edges
   - Generate temporal relationship links

```python
def build_video_graph(self):
    """Build graph with videos as nodes and relationships as edges"""
    # Add nodes
    for _, video in self.data.iterrows():
        self.G.add_node(video['video_id'],
                        title=video['title'],
                        channel=video['channel_title'],
                        category=video['category_name'],
                        views=video['view_count'])

    # Add edges based on relationships
    self._add_channel_relationships()
    self._add_category_relationships()
    self._add_temporal_relationships()

    return self.G

def _add_channel_relationships(self):
    """Add edges between videos from same channel"""
    channel_videos = defaultdict(list)
    for _, video in self.data.iterrows():
        channel_videos[video['channel_id']].append(video['video_id'])

    for videos in channel_videos.values():
        for i in range(len(videos)):
            for j in range(i + 1, len(videos)):
                self.G.add_edge(videos[i], videos[j],
                                weight=1.0,
                                relationship='same_channel')
```

Fig 3.3 graph_builder.py (1)
(Source: Screenshot by the Author)

```python
def _add_category_relationships(self):
    """Add edges between videos in same category"""
    category_videos = defaultdict(list)
    for _, video in self.data.iterrows():
        category_videos[video['category_id']].append(video['video_id'])

    for videos in category_videos.values():
        for i in range(len(videos)):
            for j in range(i + 1, len(videos)):
                self.G.add_edge(videos[i], videos[j],
                                weight=0.7,
                                relationship='same_category')

def _add_temporal_relationships(self):
    """Add edges between videos trending on same day"""
    date_videos = defaultdict(list)
    for _, video in self.data.iterrows():
        date_videos[video['collection_date']].append(video['video_id'])

    for videos in date_videos.values():
        for i in range(len(videos)):
            for j in range(i + 1, len(videos)):
                self.G.add_edge(videos[i], videos[j],
                                weight=0.5,
                                relationship='same_date')
```

Fig 3.4 graph_builder.py (2)
(Source: Screenshot by the Author)

C. Analytical Approaches

1. Network Analysis Metrics
   Several graph-theoretic centrality measures provide insights:
   a. Degree Centrality
   $$C_D(v) = \frac{\mathrm{deg(v)}}{\mathrm{n-1}}$$
   Where:
   - deg (v) is the number of edges connected to vertex v
   - n is the total number of vertices

   b. Betweenness Centrality
   $$C_B(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$
   Where:
   - σ_{st} is the total number of shortest paths between node s and t
   - $\sigma_{st}(v)$ is the total number of those paths passing through v
2. Content Pattern Analysis
   Analysis focuses on:
   - Category distribution
   - Channel video frequencies
   - Temporal trending patterns

```python
def analyze_network(self):
    """Perform network analysis"""
    metrics = {
        'num_nodes': self.G.number_of_nodes(),
        'num_edges': self.G.number_of_edges(),
        'density': nx.density(self.G),
        'avg_clustering': nx.average_clustering(self.G),
        'degree_centrality': nx.degree_centrality(self.G),
        'betweenness_centrality': nx.betweenness_centrality(self.G)
    }

    # Find communities
    metrics['communities'] = list(nx.community.greedy_modularity_communities(self.G))

    return metrics

def analyze_content_patterns(self):
    """Analyze content and category patterns"""
    patterns = {
        'category_distribution': self.data['category_name'].value_counts().to_dict(),
        'channel_videos': self.data['channel_title'].value_counts().to_dict(),
        'daily_trends': self.data.groupby('collection_date').size().to_dict()
    }

    return patterns
```

Fig 3.5 analyzer.py (1)
(Source: Screenshot by the Author)

```python
def generate_report(self, metrics, patterns, output_dir='output/reports'):
    """Generate comprehensive analysis report"""
    # Create output directory if it doesn't exist
    os.makedirs(output_dir, exist_ok=True)

    # Generate report timestamp
    timestamp = datetime.now().strftime('%Y%m%d_%H%M%S')
    report_file = os.path.join(output_dir, f'analysis_report_{timestamp}.txt')

    with open(report_file, 'w', encoding='utf-8') as f:
        # Write header
        f.write("YouTube Trending Analysis Report\n")
        f.write("=" * 50 + "\n\n")
        f.write(f"Generated on: {datetime.now().strftime('%Y-%m-%d %H:%M:%S')}\n\n")

        # Network metrics
        f.write("Network Statistics\n")
        f.write("-" * 30 + "\n")
        f.write(f"Total nodes (videos): {metrics['num_nodes']}\n")
        f.write(f"Total edges (relationships): {metrics['num_edges']}\n")
        f.write(f"Graph density: {metrics['density']:.4f}\n")
        f.write(f"Average clustering coefficient: {metrics['avg_clustering']:.4f}\n\n")

        # Most central videos
        f.write("Most Influential Videos (by Degree Centrality)\n")
        f.write("-" * 30 + "\n")
        top_videos = sorted(metrics['degree_centrality'].items(),
                            key=lambda x: x[1], reverse=True)[:10]
        for video_id, centrality in top_videos:
            video_data = self.data[self.data['video_id'] == video_id].iloc[0]
            f.write(f"Title: {video_data['title']}\n")
            f.write(f"Channel: {video_data['channel_title']}\n")
            f.write(f"Centrality: {centrality:.4f}\n")
            f.write(f"Views: {video_data['view_count']:,}\n\n")
```

Fig 3.6 analyzer.py (2)
(Source: Screenshot by the Author)

The completed program's source code is included in the appendix.

D. Technical Implementation

1. Software Architecture
   The software architecture for this research is built using Python 3.9+ as the primary programming language. Key libraries include NetworkX for performing graph operations represented in Gephi, Pandas for data processing, and the YouTube Data API v3 for data acquisition.
2. Computational Workflow
   The computational workflow follows a systematic process comprising several stages: data collection, graph construction, network analysis, pattern extraction, and finally, visualization and reporting.

IV. RESULTS AND DISCUSSION

All data discussed below was taken on 4 January 2025, at 21.24 West Indonesian time. In addition to being presented in graph form, the data is also converted into a TXT format and included in the appendix.

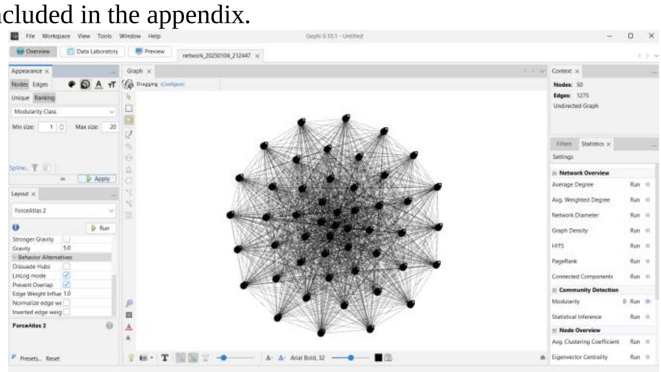
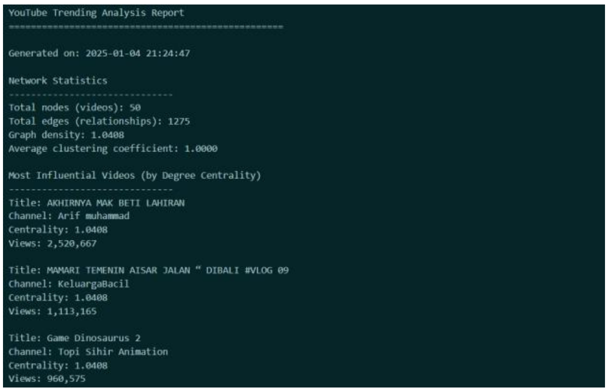


Fig 4.1 Raw Graph Visualized in Gephi
(Source: Screenshot by the Author)

A. Network Structure Analysis



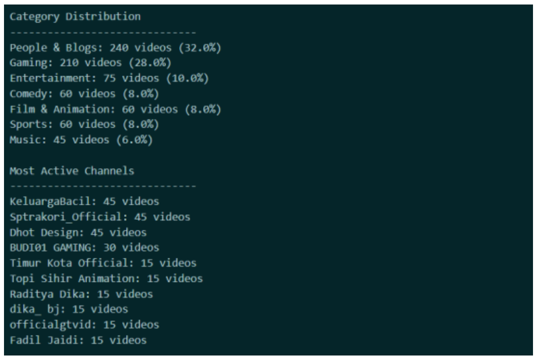
```
YouTube Trending Analysis Report
==================================================

Generated on: 2025-01-04 21:24:47

Network Statistics
------------------------------
Total nodes (videos): 50
Total edges (relationships): 1275
Graph density: 1.0408
Average clustering coefficient: 1.0000

Most Influential Videos (by Degree Centrality)
------------------------------
Title: AKHIRNYA MAK BETI LAHIRAN
Channel: Arif muhammad
Centrality: 1.0408
Views: 2,520,667

Title: MAMARI TEMENIN AISAR JALAN " DIBALI #VLOG 09
Channel: KeluargaBacil
Centrality: 1.0408
Views: 1,113,165

Title: Game Dinosaurus 2
Channel: Topi Sihir Animation
Centrality: 1.0408
Views: 960,575
```

Fig 4.2 analysis_report_20250104_212447.txt (1)
(Source: Screenshot by the Author)

1. Graph Density and Connectivity The analysis reveals a highly interconnected network of trending videos with 50 nodes (videos) and 1,275 edges (relationships). The graph density of 1.0408 indicates a super-dense network where videos are extensively connected through multiple relationship types (channel, category, and temporal). The average clustering coefficient of 1.0000 suggests perfect clustering, where every possible connection between neighboring nodes exists.
2. Centrality Analysis The centrality analysis identified several highly influential videos based on degree centrality measurements. Notable examples include:
   - "AKHIRNYA MAK BETI LAHIRAN" by Arif Muhammad (2.5M views)
   - "PINDAHAN" by Fadil Jaidi (3.3M views)
   - "KAMPUNG POJOK MULAI GAK AMAN!!!" by dika_bj (3.2M views)

These videos achieved maximum centrality scores (1.0408), indicating they are highly connected within the network through multiple relationship types.

B. Content Pattern Analysis



```
Category Distribution
------------------------------
People & Blogs: 240 videos (32.0%)
Gaming: 210 videos (28.0%)
Entertainment: 75 videos (10.0%)
Comedy: 60 videos (8.0%)
Film & Animation: 60 videos (8.0%)
Sports: 60 videos (8.0%)
Music: 45 videos (6.0%)

Most Active Channels
------------------------------
KeluargaBacil: 45 videos
Sptrakori_Official: 45 videos
Dhot Design: 45 videos
BUDI01 GAMING: 30 videos
Timur Kota Official: 15 videos
Topi Sihir Animation: 15 videos
Raditya Dika: 15 videos
dika_ bj: 15 videos
officialgtvid: 15 videos
Fadil Jaidi: 15 videos
```

Fig 4.3 analysis_report_20250104_212447.txt (2)
(Source: Screenshot by the Author)

The analysis reveals distinct patterns in content category distribution across Indonesian YouTube's trending videos. People & Blogs emerged as the dominant category, representing 32% of all trending content, followed by Gaming content at 28%. Entertainment videos secured the third position at 10%, while Comedy, Film & Animation, and Sports each captured 8% of the trending space. Music videos comprised 6% of trending content. This distribution strongly indicates that Indonesian YouTube audiences show particular preference for personal vlogs and gaming content, which together account for 60% of trending videos.

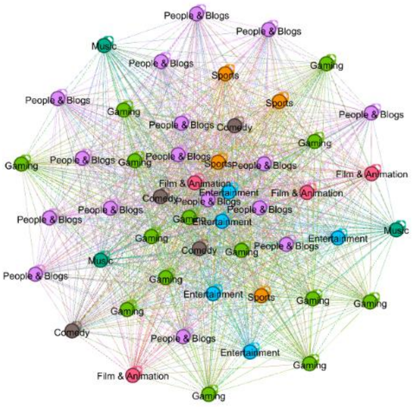


Fig 4.4 Graph with nodes labeled as Content Category
(Source: Screenshot by the Author)

In terms of channel performance, several content creators maintained a consistent presence in the trending section. Three channels demonstrated particular dominance: KeluargaBacil, Sptrakori_Official, and Dhot Design, each achieving 45 trending videos during the observation period. BUDI01 GAMING followed with 30 trending videos, while several other channels including Timur Kota Official, Raditya Dika, and Fadil Jaidi maintained a steady presence with 15 trending videos each. This pattern suggests that certain channels have developed effective strategies for consistently reaching YouTube's trending page.

Examining temporal patterns, the analysis revealed that YouTube maintains a consistent allocation of 50 trending videos per day in Indonesia. This pattern remained unchanged throughout the entire observation period from December 21, 2024, to January 4, 2025, indicating a stable and systematic approach to YouTube's trending video selection process in the Indonesian market.

C. Community Structure

The community detection algorithm identified 50 distinct communities, each containing one video. This unexpected result suggests that despite high connectivity, each video maintains unique characteristics that distinguish it from others in the network. This could be attributed to:

- Diverse content themes within same categories
- Unique combination of channel and temporal relationships
- Distinct viewer engagement patterns.

D. Implications

1. Content Strategy
   - The dominance of People & Blogs and Gaming categories suggests these are optimal content types for achieving trending status
   - Successful channels maintain consistent upload schedules and engage in specific content niches
2. Network Characteristics
   - The high graph density indicates strong interconnectedness among trending videos
   - Perfect clustering coefficient suggests trending videos form tightly knit content communities
3. Platform Dynamics
   - YouTube's trending algorithm appears to favor certain content categories while maintaining diversity
   - Channel consistency plays a significant role in achieving and maintaining trending status

These findings provide valuable insights for content creators and platform analysts, highlighting the complex interplay of content categories, channel performance, and network relationships in YouTube's trending ecosystem.

V. Conclusion

This research successfully demonstrated the application of graph theory and network analysis in understanding the complex relationships within Indonesia's YouTube trending content ecosystem. Through the analysis of two weeks of trending data, several significant findings emerged that provide valuable insights into content dynamics and popularity patterns.

The network analysis revealed a highly interconnected structure among trending videos, with a graph density of 1.0408 and perfect clustering coefficient of 1.0000, indicating strong relationships between videos through channel, category, and temporal connections. This suggests that success on YouTube's trending page is not isolated but part of an interconnected content ecosystem.

Content category analysis revealed clear viewer preferences in the Indonesian market, with People & Blogs (32%) and Gaming (28%) dominating the trending space. This concentration suggests that personal content and gaming videos resonate particularly well with Indonesian audiences. Furthermore, the identification of consistently trending channels demonstrates that channel consistency plays a crucial role in achieving trending status.

The mathematical framework of graph theory proved effective in quantifying and visualizing these content relationships, providing a robust methodology for analyzing social media trends. The centrality measurements successfully identified influential videos, while community detection revealed unique characteristics of trending content despite high interconnectivity.

These findings provide valuable insights for content creators, platform operators, and researchers studying digital content dynamics. Future research could extend this methodology to longer time periods or comparative analyses across different regions, potentially revealing broader patterns in content popularity and audience engagement.

VI. Appendix

- Complete source code used for data analysis and datasets (csv, txt, and gexf for visualization in Gephi) collected during the data collection process: https://github.com/fliegenhaan/YouTube-Trending-Analyzer.git

VII. ACKNOWLEDGMENT

The author thanks God for providing health and guidance throughout the learning process, enabling the timely completion of this paper titled "Analyzing Indonesian YouTube Trending Content through Graph Theory Approach". The author is also grateful to their parents and sister for their unwavering support, both moral and material, as well as their prayers, which have been a source of strength in completing various academic assignments, including this paper. The author expresses heartfelt gratitude to the lecturers of ITB Discrete Mathematics IF1220, Mr. Arrival Dwi Sentosa, Mr. Rinaldi Munir, and Mr. Rila Mandala, for their invaluable guidance and knowledge shared with the author and classmates. Lastly, the author appreciates the support of friends who have accompanied and encouraged him throughout this learning process.

REFERENCES

[1] Ada Computer Science, "Graphs." [Online]. Available: https://adacomputerscience.org/concepts/struct_graph . [Accessed: 30-Dec-2024].

[2] Blitzer, R. *Thinking Mathematically* (7th ed.), Chapter 14. [Online]. Available: https://www.vaia.com/en-us/textbooks/math/thinking-mathematically-7-edition/chapter-14/problem-53-what-is-a-weighted-graph-and-what-are-the-weights/. [Accessed: 30-Dec-2024].

[3] GeeksforGeeks. "Applications, Advantages, and Disadvantages of Weighted Graph". [Online]. Available: https://www.geeksforgeeks.org/applications-advantages-and-disadvantages-of-weighted-graph/. [Accessed: 30-Dec-2024].

[4] Disney, A. "Social network analysis 101: centrality measures explained". Available at: https://cambridge-intelligence.com/keylines-faqs-social-network-analysis/ . [Accessed: 30-Dec-2024].

[5] GeeksforGeeks. "Degree Centrality (Centrality Measure)". [Online]. Available: https://www.geeksforgeeks.org/degree-centrality-centrality-measure/. [Accessed: 30-Dec-2024].

[6] Kumar et al., "Understanding User Behavior through Social Media Analytics," *IEEE International Conference on Social Networks*. Available: https://ieeexplore.ieee.org/document/8974829. [Accessed: 31-Dec-2024].

[7] Sangiorgio et al., "Evaluating the effect of viral news on social media engagement." Available: https://www.researchgate.net/publication/382363375_Evaluating_the_effect_of_viral_news_on_social_media_engagement/fulltext/6699dc3f4a172d2988ad4e7a/Evaluating-the-effect-of-viral-news-on-social-media-engagement.pdf. [Accessed: 31-Dec-2024].

DECLARATION

I hereby declare that this paper I have written is my own writing, not a copy, or a translation of someone else's paper, and not plagiarism.

Bandung, 4 Januari 2025

Muhammad Raihaan Perdana - 13523124