

ANALISIS AKTOR POPULAR DAN SUTRADARA BERPENGARUH BERDASARKAN DATA DBPEDIA MENGUNAKAN ALGORITMA CLOSENESS CENTRALITY DAN NODE2VEC

KARIMA MUFIDAH¹, NOVYANT SYAHPUTRA², NUR AINI RAKHMAWATI³
Institut Teknologi Sepuluh Nopember¹²³
e-mail : karimamufidah@gmail.com

ABSTRACT

The rapid development of web technology makes it easier for humans to access various information. The third generation of web-based internet services (Web 3.0) has introduced the Semantic Web which aims to enable content on the web to be understood by computers. The application of semantic web can be done to retrieve a dataset from Dbpedia Indonesia, which is a list of Indonesian films, for further data analysis. The purpose of data analysis is to find out the most popular actors and directors in the Indonesian film industry. This study uses closeness centrality and Node2vec algorithms to determine the level of popularity of actors. In addition, this study also uses density graph to determine the influential directors in the Indonesian film industry. The results of the algorithm calculations are visualized using Neo4j, Networkx, and tSNE which are graphs. In this study it was found that Rima Melati was the most popular actor because of the highest closeness centrality value. This can also be interpreted that Rima Melati is an actor who starred in the most films. While in density graph calculation, Sophan Sophiaan is the most influential director because he directs the most movie titles.

Key Words : closeness centrality, dbpedia, density graph, networkx, node2vec

ABSTRAK

Perkembangan teknologi web yang semakin pesat membuat manusia semakin mudah dalam mengakses berbagai informasi. Generasi ketiga dari layanan internet berbasis web (Web 3.0) telah memperkenalkan Semantic Web yang bertujuan untuk memungkinkan konten pada web agar dapat dipahami oleh komputer. Penerapan Semantic Web dapat dilakukan untuk mengambil dataset dari DBpedia Indonesia, yaitu daftar film Indonesia, untuk selanjutnya dilakukan analisis data. Tujuan analisis data adalah untuk mengetahui aktor dan sutradara terpopuler dalam industri perfilman Indonesia. Penelitian ini menggunakan algoritma closeness centrality dan Node2vec untuk menentukan tingkat popularitas aktor. Selain itu, penelitian ini juga menggunakan density graph untuk mengetahui sutradara yang berpengaruh di industri perfilman Indonesia. Hasil dari perhitungan algoritma tersebut divisualisasikan menggunakan Neo4j, Networkx dan tSNE yang mana berupa graf. Pada penelitian ini ditemukan bahwa Rima Melati merupakan aktor terpopuler karena nilai closeness centralitynya tertinggi. Hal ini juga dapat diartikan bahwa Rima Melati merupakan aktor yang membintangi judul film terbanyak. Sedangkan pada perhitungan density graph, Sophan Sophiaan merupakan sutradara yang paling berpengaruh karena menyutradarai paling banyak judul film.

Kata Kunci: closeness centrality, dbpedia, density graph, networkx, node2vec

PENDAHULUAN

Internet telah menjadi sarana untuk mempermudah setiap orang mengakses segala bentuk pengetahuan dan informasi. Hal ini dimungkinkan karena adanya perkembangan teknologi web dari tahun ke tahun [1]. Pada tahun 2002, Tim Berners-Lee mencetuskan teknologi web terbaru yaitu *Semantic Web* yang dinamai dengan era Web 3.0. Hadirnya *semantic web* bertujuan untuk memperkaya informasi yang diberikan sehingga menjadi lebih baik dalam pendefinisian, agar memungkinkan komputer dapat memahami informasi yang telah diberikan sehingga komputer dan manusia dapat bekerja sama [2]. Cara kerja *semantic web* adalah dengan mengambil URI (*Uniform Resource Identifier*) pada sebuah web.

Pengaplikasian *semantic web* dapat dilakukan untuk pengambilan data pada DBpedia Indonesia. DBpedia merupakan sebuah komunitas yang bergerak untuk mengekstrak informasi terstruktur dari Wikipedia, dalam hal ini merupakan Wikipedia Indonesia, dan menyediakan informasi tersebut dalam sebuah web [1]. Untuk mengakses pengetahuan (*knowledge*) DBpedia, digunakan URI *resource* dari entitas dalam bentuk SPARQL.

Adapun beberapa penelitian sebelumnya yang terkait dengan penelitian ini yaitu Adang Rochiyat dan Arief Wibowo yang melakukan analisis aktor berpengaruh dan aktor populer dengan metode *degree centrality* dan *follower rank* dengan dataset berasal dari tagar Twitter #gejayanmemanggil [3]. Dari penelitian tersebut diketahui bahwa aktor Ryan Restu merupakan aktor yang berpengaruh karena memiliki nilai *degree centrality* tertinggi dan aktor Tirto ID merupakan aktor yang paling populer karena memiliki nilai *follower rank* yang tertinggi. Selain itu, beberapa penelitian sebelumnya [4] [5] [6] [7] juga telah membuktikan bahwa dengan menggunakan algoritma *closeness centrality* dapat menunjukkan nilai sentralitas yang berfungsi untuk mengetahui tingkat popularitas dan pengaruh *node* (aktor) pada sebuah jaringan.

Pada penelitian ini, data yang diambil pada DBpedia Indonesia adalah daftar film Indonesia yang nantinya akan dilakukan analisis data. Tujuan dari analisis data adalah untuk mengetahui aktor dan sutradara terpopuler dalam industri perfilman Indonesia. Selain itu, analisis ini juga bertujuan untuk mengetahui siapa tokoh (aktor/sutradara) yang berpengaruh di industri tersebut. Metode yang digunakan adalah algoritma *closeness centrality* dan Node2vec untuk mengetahui tingkat popularitas artis dan sutradara. Hasil perhitungan algoritma tersebut divisualisasikan dengan Neo4j, Networkx, dan tSNE yang mana berupa *graph*. Penelitian ini diharapkan dapat memberikan wawasan dan pengetahuan baru kepada masyarakat dan pengamat film mengenai aktor dan sutradara yang memiliki

pengaruh di industri perfilman Indonesia.

TINJAUAN PUSTAKA

Penelitian ini menggunakan dasar teori yang menjadi acuan selama pengerjaan sehingga dapat lebih sistematis.

SPARQL

Simple Protocol and RDF Query Language atau lebih dikenal dengan SPARQL adalah bahasa yang digunakan untuk melakukan *query* data dalam *semantic web* [8]. Dengan adanya SPARQL, masing-masing sumber data dapat terhubung satu sama lain. Dalam penggunaannya, *query* SPARQL memiliki struktur yang umumnya digunakan seperti contoh yang ditunjukkan pada Gambar 1.

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name
WHERE {
  <http://example.org/alice#me> foaf:knows [ foaf:name ?name ] .
  SERVICE <http://dbpedia.org/sparql>
    { <http://dbpedia.org/resource/Snoopy> foaf:name ?name }
```

Gambar 1. Struktur Query SPARQL

Gambar 1 menunjukkan bahwa umumnya pada *query* SPARQL terdapat PREFIX, SELECT, dan WHERE. PREFIX berfungsi untuk mempersingkat penulisan alamat data sehingga penulisan *query* dapat dilakukan lebih mudah. SELECT berfungsi untuk mengambil data dari SPARQL *endpoint* dan hasilnya berupa tabel, sedangkan WHERE merupakan syarat atau permintaan yang telah ditentukan. Pada contoh tersebut, *output* yang dihasilkan berupa nama dari *dbpedia resource* yang digunakan, yaitu Snoopy, yang ditunjukkan pada Gambar 2.

```
?name
"Snoopy"@en
```

Gambar 2. Contoh Output Query SPARQL

Node2vec

Node2vec merupakan sebuah algoritma *framework* yang berfungsi untuk memetakan *node* ke dalam bentuk *vector* pada sebuah jaringan [9]. Dengan memilih satu titik/*node* yang tepat pada lingkungan sekitarnya (jaringan), Node2vec dapat mempelajari representasi yang mengatur *node* berdasarkan pada peran jaringan atau komunitas tempat *node* tersebut bergabung.

Node2vec memiliki parameter p dan q yang berfungsi untuk menampilkan variabel bias *random walks*. Parameter p atau disebut juga dengan *return parameter* berfungsi untuk mengontrol kemung-

kinan perjalanan kembali sebuah *node* dalam jaringannya. Sedangkan parameter *q* atau disebut juga dengan *in-out parameter* berfungsi memungkinkan sebuah *node* melakukan pencarian untuk membedakan antara *node* ke dalam (*inward*) dan ke luar (*outward*).

Closeness Centrality

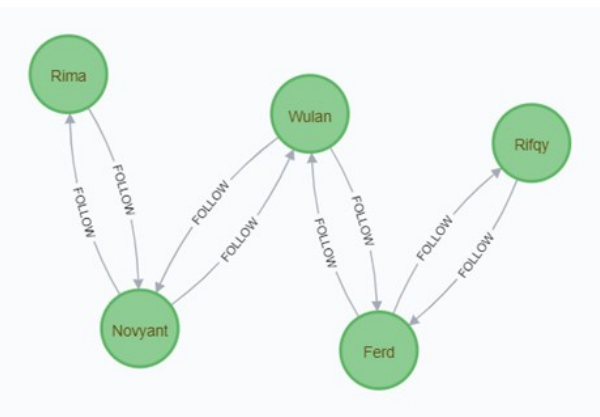
Closeness centrality merupakan algoritma yang berfungsi untuk menghitung jarak rata-rata *node* dengan semua *node* lain dalam suatu jaringan [10]. Ukuran ini menggambarkan kedekatan satu *node* dengan *node* yang lain dalam suatu jaringan. Semakin besar nilai *closeness centrality* yang dimiliki menandakan bahwa *node* tersebut memiliki tingkat eksistensi paling tinggi diantara *node* yang lain. Gambar 3 menunjukkan contoh graf *closeness centrality*.

Dari contoh graf yang ditunjukkan oleh Gambar 3, nilai *closeness centrality* tertinggi dimiliki oleh Wulan karena menghubungkan banyak *node* dalam jaringan.

Untuk menghitung nilai *closeness centrality* pada sebuah *node* dapat menggunakan rumus (1) berikut.

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

Pada rumus tersebut, *u* melambangkan sebuah *node*, *n* merupakan jumlah *node* dalam komponen yang sama (*subgraph* atau *graph*) seperti *u*, sedangkan *d(u,v)* adalah jarak jalur terpendek antara *node v* dan *node (u)* lainnya.



Gambar 3. Contoh Graf Closeness Centrality

Neo4j

Neo4j merupakan *graph database* yang diperkenalkan pada tahun 2007. Pada Neo4j, graf memiliki properti yang terdiri dari *node*, *relationship*, *direction*, dan *label* [11]. *Node* merupakan aktor yang memiliki peran dalam jaringan, *relationship* merupakan hubungan yang dimiliki antar *node*, *direction* merupakan arah hubungan dari *node* satu ke *node* yang lain, sedangkan *label* digunakan untuk memberi nama pada *node* dan *relationship*.

Networkx

Networkx adalah salah satu *package* pada bahasa pemrograman Python yang berfungsi untuk mengeksplorasi dan menganalisis jaringan dan algoritma jaringan [12]. Networkx menyediakan berbagai jenis graf untuk menggambarkan jaringan, termasuk *simple graph* dan *directed graph*. Beberapa contoh algoritma graf yang dapat diimplementasikan dengan Networkx adalah *shortest paths*, *betweenness centrality*, *closeness centrality*, dan lain sebagainya.

t-SNE

t-SNE juga merupakan salah satu *package* pada Python yang berfungsi untuk membuat visualisasi data [13]. Dengan t-SNE, visualisasi yang dihasilkan jauh lebih baik karena algoritma ini mengurangi kecenderungan titik kerumunan di tengah peta. t-SNE lebih baik daripada teknik yang ada saat membuat peta tunggal (*single map*) yang mengungkapkan struktur pada banyak skala yang berbeda.

Density Graph

Density didefinisikan sebagai rasio antara *edge* (tepi) dan *node* (simpul) dari sebuah *subgraph* atau rata-rata *degree* (*node* yang terhubung) dibagi 2 [14]. Dengan *density graph* dapat diketahui jarak antara *node* satu dengan yang lain. *Density graph* juga dapat menentukan *node* mana yang memiliki pengaruh terbesar dalam sebuah jaringan.

METODE

Penelitian ini dilakukan sesuai dengan alur yang ditunjukkan pada Gambar 4. Tahapan pada

alur tersebut meliputi *query* SPARQL, pemodelan *graph*, perhitungan algoritma *closeness centrality* dan *Node2vec*, dan yang terakhir adalah visualisasi hasil dari algoritma tersebut.



Gambar 4. Alur Metode Penelitian

Query SPARQL

Query yang digunakan untuk mendapatkan data daftar film dari DBpedia Indonesia [15], digunakan *property* *wikiPageWikiLink* yang dituliskan dengan *dbpedia-owl:wikiPageWikiLink*. Untuk mengambil data sutradara digunakan *property* *director* yang dituliskan dengan *dbpedia-owl:starring*. Detil dari *query* adalah sebagai berikut:

```

SELECT ?film ?labelFilm ?director ?
labelDirector ?starring ?labelStarring
WHERE{
  <http://id.dbpedia.org/resource/
Daftar_film_Indonesia> dbpedia-
owl:wikiPageWikiLink ?film.
  ?film rdfs:label ?labelFilm.
  ?film dbpedia-owl:director ?director.
  ?director rdfs:label ?labelDirector.
  ?film dbpedia-owl:starring ?starring.
  ?starring rdfs:label ?labelStarring.
}
  
```

Data yang dihasilkan dari *query* tersebut kemudian disimpan dalam format CSV. Tabel 1 merupakan sampel data hasil *query*.

Tabel 1. Sampel Data Hasil Query

| No | Film | Sutradara | Aktor |
|----|--------------------------|----------------|----------------|
| 1. | Pengkhianatan G 30 S/PKI | Arifin C. Noer | Umar Kayam |
| 2. | Pengkhianatan G 30 S/PKI | Arifin C. Noer | Ade Irawan |
| 3. | Pengkhianatan G 30 S/PKI | Arifin C. Noer | Sofia WD |
| 4. | Taksi (film) | Arifin C. Noer | Rano Karno |
| 5. | Taksi (film) | Arifin C. Noer | Meriam Bellina |

Dari *query* SPARQL yang telah dijalankan, dihasilkan sejumlah 3874 baris data yang terdiri dari 3 kolom, yaitu film, sutradara, dan pemain (aktor). Statistik jumlah proporsi data ditunjukkan oleh tabel 2. Data tersebut disimpan pada repositori Github yang terhubung dengan Zenodo [16]. Data ini dihasilkan pada saat *query* bulan Mei 2020.

Tabel 2. Statistik Jumlah Data

| Node | Jumlah |
|------------|--------|
| Judul Film | 611 |
| Sutradara | 152 |
| Pemain | 1006 |

Pemodelan Graph

Setelah data didapatkan menggunakan *query* SPARQL selanjutnya dilakukan pemodelan *graph*. Pemodelan *graph* merupakan proses mengubah bentuk data yang awalnya berbasis tabel menjadi graf. Graf yang akan dibuat memiliki 3 jenis *node* yang berasal dari 3 kolom pada Tabel 1, yaitu *node* judul film, sutradara, dan aktor. Proses ini dilakukan dengan menggunakan aplikasi Neo4j. Dengan adanya pemodelan graf dapat diketahui *network* yang terbentuk dari *node-node* di dalamnya.

Menjalankan Node2vec

Algoritma Node2vec akan dijalankan dengan bahasa pemrograman Python menggunakan aplikasi Jupyter Notebook untuk menghasilkan angka *perplexity* dan visualisasi dari graf. *Node* yang digunakan dalam algoritma ini adalah judul film dengan parameter 'film', nama sutradara dengan parameter 'sutradara', dan nama aktor dengan parameter 'pemain'.

Algoritma Closeness Centrality

Algoritma *closeness centrality* akan dijalankan pada aplikasi Neo4j. Pada algoritma ini terdapat beberapa parameter yang digunakan dalam penelitian, Nama dan fungsi dari parameter ditunjukkan pada Tabel 3.

Tabel 3. Parameter yang Digunakan

| Nama | Tipe | Fungsi |
|--------------|---------|---|
| Label | String | Memberi label pada <i>node</i> graf |
| Relationship | String | Memberi relasi antar <i>node</i> pada graf |
| Concurrency | Integer | Menjalankan algoritma yang diinginkan |
| Graph | String | Menampilkan graf dengan label dan <i>relationship</i> |

Visualisasi

Setelah melakukan perhitungan algoritma, hasil yang diperoleh akan divisualisasikan menggunakan bahasa pemrograman Python dengan *library* Networkx dan t-SNE. Melalui visualisasi ini dapat diketahui lebih jelas siapa aktor dan sutradara yang populer dan berpengaruh.

HASIL DAN PEMBAHASAN

Pemodelan Graph menggunakan Neo4j

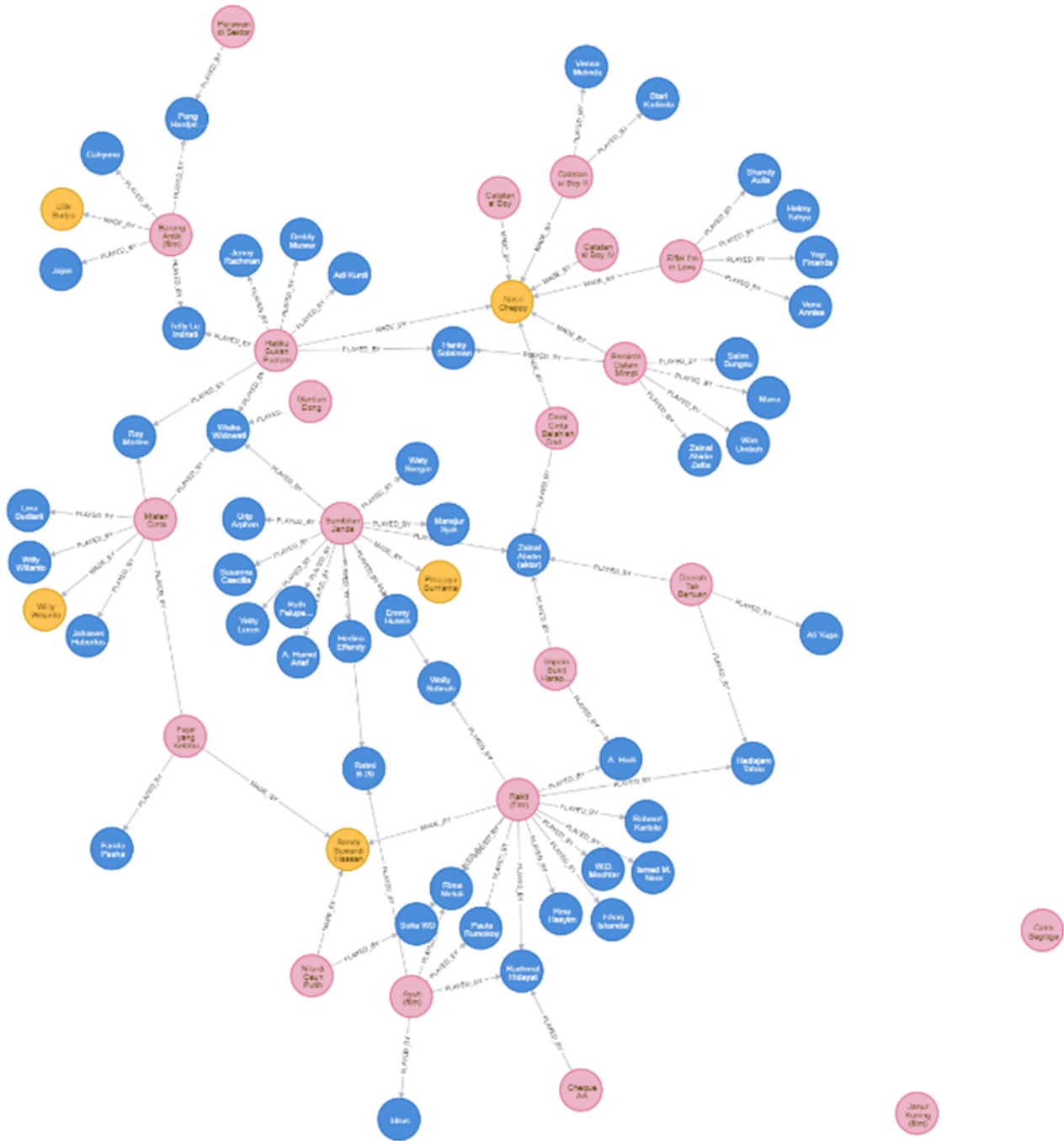
Untuk membuat pemodelan graf dari data daftar film Indonesia yang telah didapatkan harus dilakukan pemisahan data sutradara pada tabel sebelumnya dan digantikan oleh id sutradara. Pemberian id pada sutradara harus dilakukan agar tidak ada data yang terduplikasi pada saat membuat model graf. Setelah itu data dapat diimpor ke aplikasi Neo4j. Gambar 5 menunjukkan pemodelan graf yang dihasilkan dengan pemberian warna yang berbeda berdasarkan atribut *node*. Atribut judul film menggunakan warna merah muda, atribut aktor menggunakan warna biru, dan untuk atribut sutradara menggunakan warna kuning. Begitu juga dengan jenis *relationship*-nya. Jenis *relationship* antara judul film dan sutradara pada graf tersebut adalah MADE_BY (dibuat oleh), sedangkan jenis *relationship* antara judul film dan aktor pada graf adalah PLAYED_BY (dibintangi oleh).

Perhitungan Algoritma Closeness Centrality

Saat perhitungan algoritma *closeness centrality*, *node* yang digunakan merupakan *node* aktor karena tujuannya adalah untuk mengetahui siapa aktor yang paling populer. Perhitungan ini juga dilakukan menggunakan aplikasi Neo4j. Hasil nilai *closeness centrality* dapat dilihat pada Tabel 4.

Tabel 4. Nilai Closeness Centrality pada Node

| Node (Aktor) | Closeness Centrality |
|----------------------------|----------------------|
| Rima Melati | 0.2593026662760035 |
| Rina Hasyim | 0.2546029919447641 |
| Nani Widjaja | 0.2523524379811805 |
| Rachmat Hidayat | 0.25134904856574836 |
| ... | ... |
| Iqbaal Dhiafakhri Ramadhan | 0.13445761166818596 |
| Teuku Rizky Muhammad | 0.13445761166818596 |
| Bastian Bintang Simbolon | 0.13445761166818596 |
| Fachri Albar | 0.13248502994011976 |



Gambar 5. Pemodelan Graf

Berdasarkan Tabel 4 didapatkan urutan *node* dengan nilai *closeness centrality* mulai dari yang tertinggi hingga yang terendah. Nilai tertinggi dimiliki oleh aktor Rima Melati dengan nilai 0.259, hal ini dapat diartikan bahwa aktor Rima Melati

yang ada pada dataset ini paling banyak memainkan judul film. Sedangkan nilai *closeness centrality* terendah dimiliki oleh aktor Fachri Albar dengan nilai 0.132 yang juga dapat diartikan bahwa aktor Fachri Albar memainkan paling sedikit judul film.

Perhitungan Algoritma Node2vec

Sebelum diproses, data yang akan dijadikan *node* dibentuk menjadi sebuah graf terlebih dahulu dengan *library* Networkx. Setelah itu algoritma Node2vec dijalankan dengan parameter *dimension* sebesar 20, *walk_length* sebesar 16, *num_walks* sebesar 100 dan *Workers* sebesar 2.

```

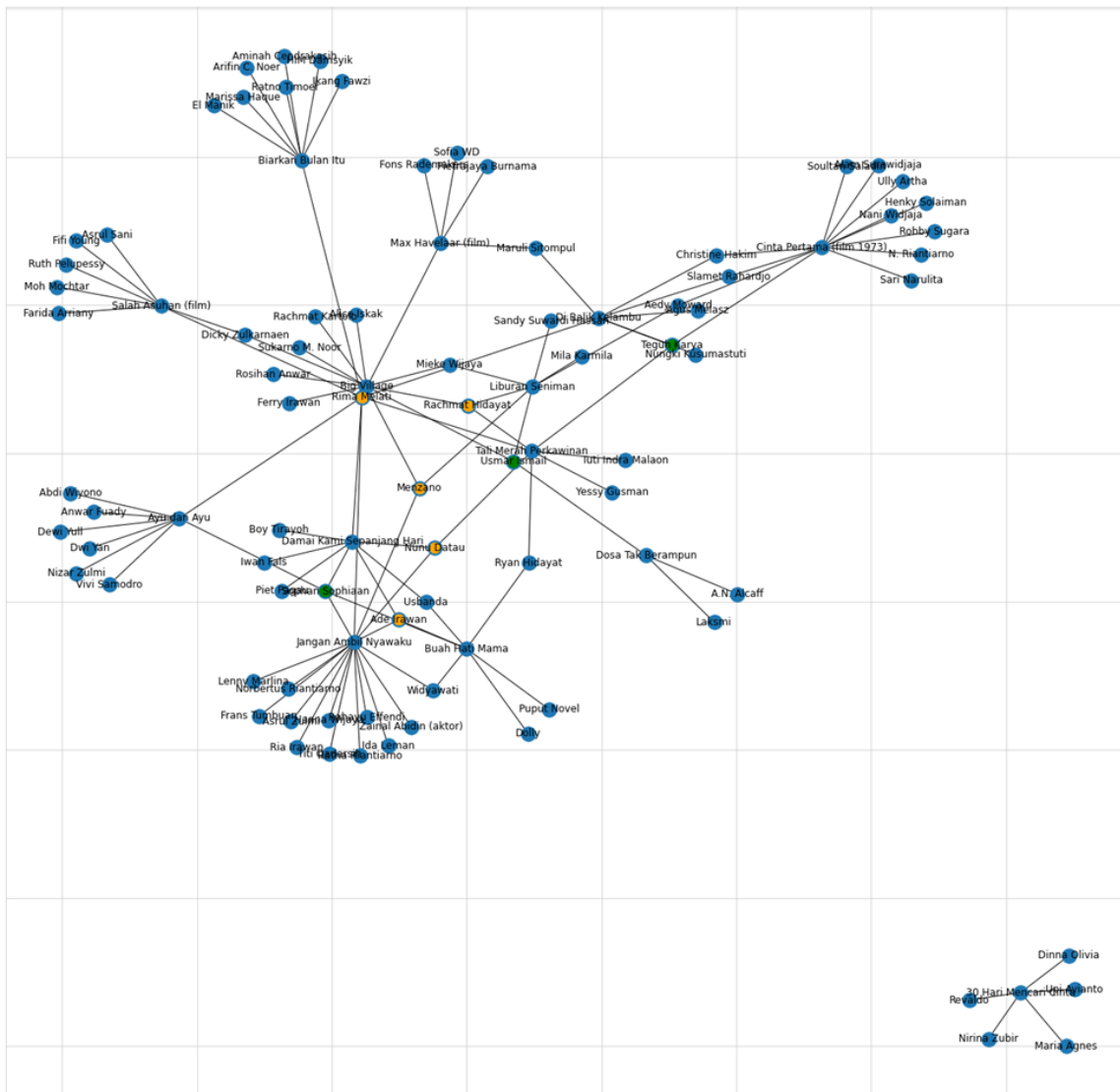
g = nx.from_pandas_edgelist(df,
source='labelFilm',
target='labelDirector')

node2vec = Node2Vec(g, dimensions=20,
walk_length=16, num_walks=100, workers=2)

model = node2vec.fit(window=10,
min_count=1)
    
```

Untuk analisa lebih lanjut, digunakan visualisasi dengan *library* Networkx pada Python, pada visualisasi ini dapat dilihat aktor populer atau banyak bermain pada sebuah judul film dan sutradara yang populer atau banyak menyutradarai judul film. Hasil dari kode tersebut ditunjukkan pada Gambar 7 dengan *node* berwarna oranye menunjukkan aktor yang memainkan 2 atau lebih relasi dengan *node* lainnya. Kemudian *node* yang berwarna hijau merupakan sutradara film yang memiliki 2 atau lebih relasi dengan *node* lain.

Gambar 6. Mengubah Data Menjadi Graf dengan Neode2vec



Gambar 7. Visualisasi Menggunakan Networkx

Dari Gambar 7 dapat diketahui bahwa Rima Melati merupakan aktor populer karena nodenya berwarna kuning. Hal ini juga sesuai dengan hasil perhitungan closeness centrality yang mana Rima Melati juga memiliki nilai tertinggi yang menandakan aktor tersebut populer. Untuk sutradara, Sophan Sophiaan merupakan sutradara berpengaruh karena menyutradarai paling banyak film. Hal ini sesuai dengan warna node Sophan Sophiaan pada graf yang berwarna hijau.

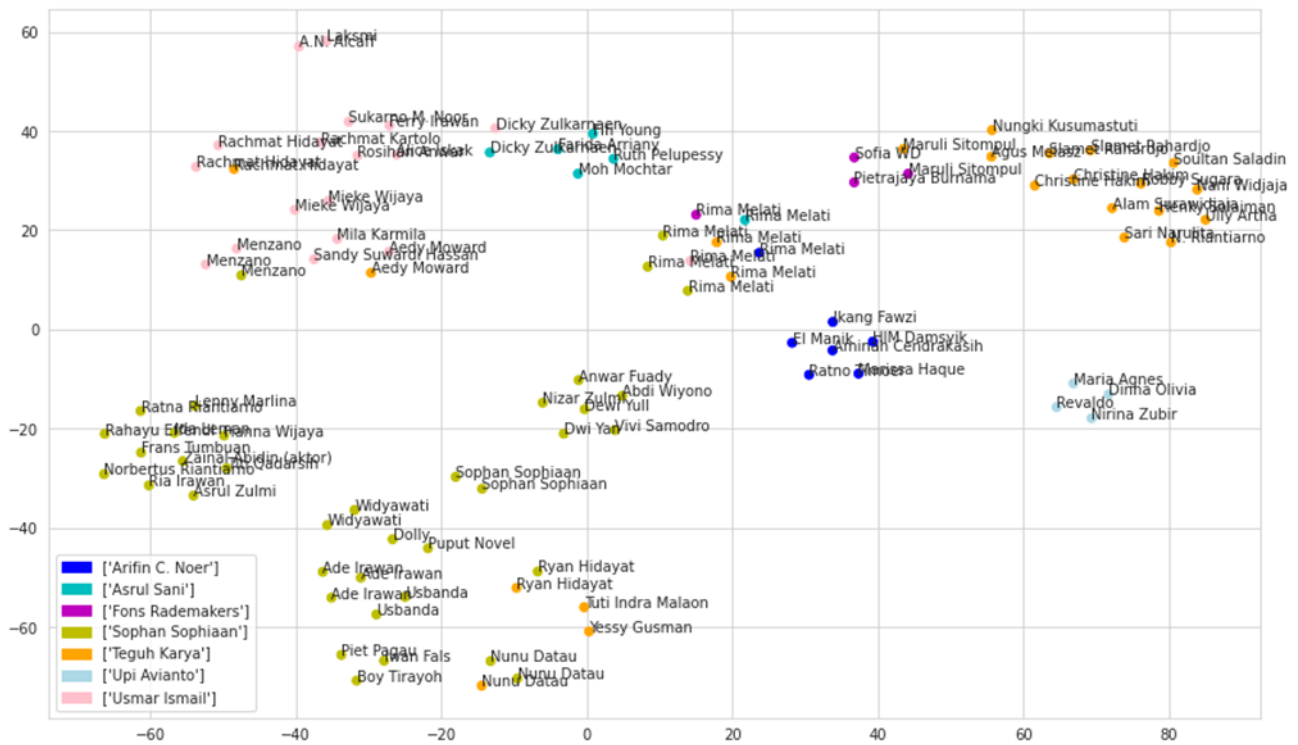
Visualisasi Data dengan t-SNE

Pada visualisasi data menggunakan t-SNE, nantinya dapat dilihat hubungan antar node. Parameter yang digunakan pada penelitian yaitu `n_components = 2` yaitu `labelDirector` dan `labelStarring`, `random_state = 7` dan `perplexity = 15`.

```
tsne = TSNE(n_components=2,
random_state=7, perplexity=15)
embeddings_2d = tsne.fit_transform
```

Gambar 8. Baris Kode Visualisasi t-SNE

Legend yang digunakan pada visualisasi t-SNE sutradara terhadap judul film adalah nama sutradara film yang diberi tanda dengan beberapa warna yang berbeda dan node nama film divisualisasikan dalam bentuk scatter plot dimana titik-titik dengan label nama film merupakan representasi judul film yang disutradarai oleh sutradara film. Grafik scatter plot ini berguna untuk mengetahui persebaran judul film dengan sutradara tertentu. Sehingga, dapat ditarik kesimpulan sutradara Sophan Sophiaan menyutradarai judul film paling banyak dan sutradara Arifin C. Noer memiliki menyutradarai judul film paling sedikit.



Gambar 9. Visualisasi t-SNE Sutradara Terhadap Judul Film

Selanjutnya, pada Gambar 10 menunjukkan visualisasi aktor terhadap judul film. *Legend* yang digunakan pada visualisasi ini adalah Judul film yang diberi tanda dengan beberapa warna berbeda berbeda dan *node* nama aktor divisualisasikan dalam bentuk *scatter plot*. Grafik *scatter plot* ini berguna untuk mengetahui persebaran aktor dengan judul film tertentu. Berdasarkan Gambar 10 dapat dilihat bahwa judul film yang dimainkan oleh aktor Rima Melati merupakan yang terbanyak. Hal ini juga mendukung pernyataan hasil perhitungan yang telah dilakukan sebelumnya yang menyatakan bahwa Rima Melati merupakan aktor yang paling populer.

besar *degree* nya. *Node* yang memiliki ukuran terbesar adalah *node* judul film Jangan Ambil Nyawaku yang menandakan bahwa di film tersebut banyak melibatkan aktor. Namun dalam penelitian ini yang ingin diketahui adalah aktor dan sutradara yang berpengaruh. Sehingga jika dilihat dengan atribut aktor, Rima Melati memiliki ukuran *node* terbesar dibandingkan dengan aktor-aktor lainnya. Sedangkan jika dilihat dengan atribut sutradara, Sophan Sophiaan juga memiliki ukuran *node* terbesar dibandingkan dengan sutradara lainnya. Hasil yang diperoleh dengan *density graph* semakin memperkuat hasil perhitungan yang dilakukan sebelumnya.

Perhitungan dan Visualisasi dengan Density Graph

Pada Gambar 11 menunjukkan *degree graph* dimana semakin besar *node* nya maka semakin

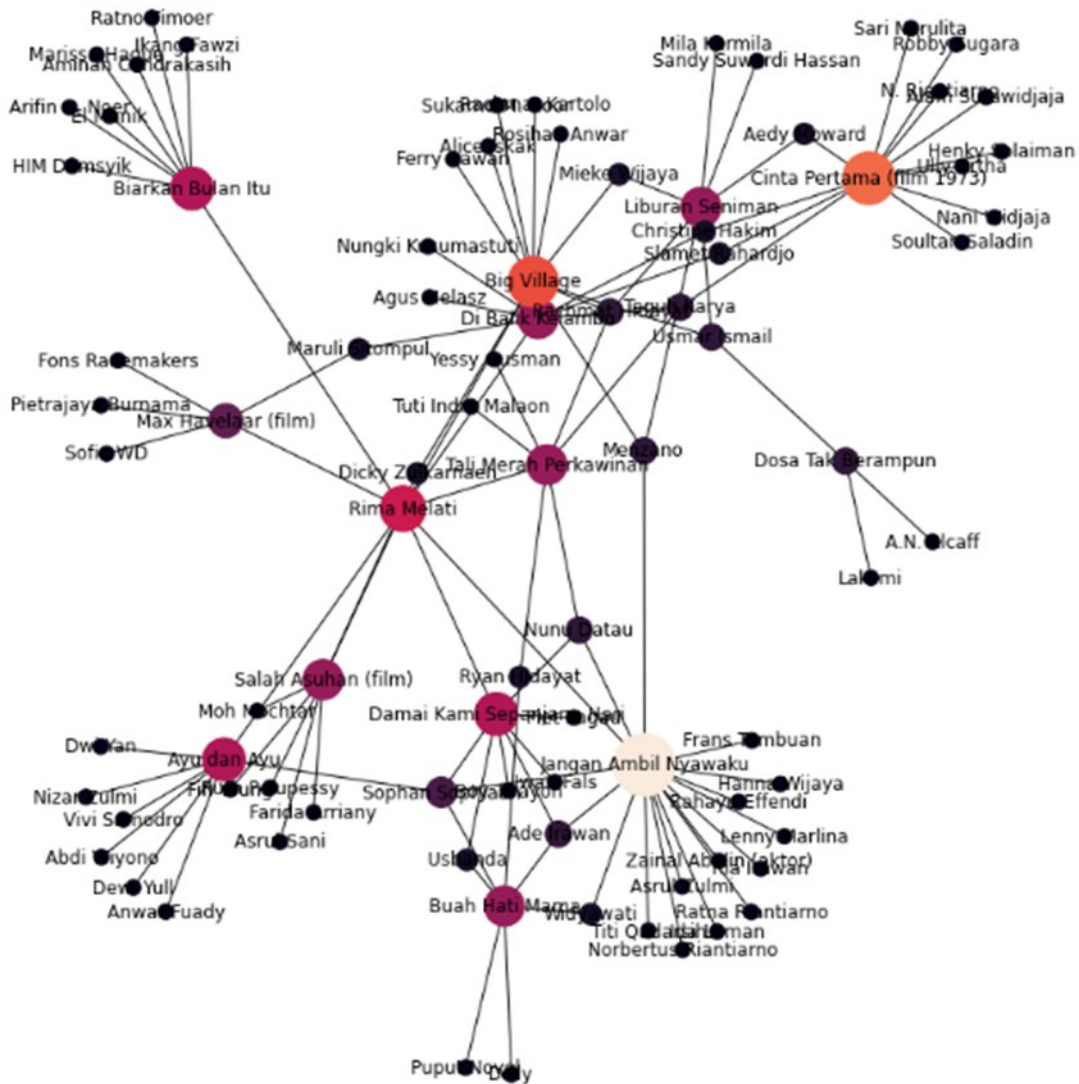


Gambar 10. Visualisasi t-SNE Aktor Terhadap Judul Film

KESIMPULAN

Berdasarkan hasil nilai *closeness centrality*, *Networkx*, *t-SNE*, dan *density graph* dapat ditarik kesimpulan bahwa Rima Melati merupakan aktor yang paling populer dalam industri perfilman Indone-

sia. Tingkat popularitas Rima Melati ditunjukkan dengan banyaknya judul film yang ia bintang. Sedangkan untuk sutradara berpengaruh dalam industri perfilman Indonesia jatuh kepada Sophan Sophiaan. hal ini dibuktikan dengan banyaknya judul film yang disutradarai olehnya.



Gambar 11. Visualisasi dengan *Density Graph*

DAFTAR PUSTAKA

- [1] A. Nurwahdah, "Sistem Tanya Jawab dengan Web Semantik," in *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*, 2015, vol. 1, no. 1.
- [2] Y. F. Badron, F. Agus, and H. R. Hatta, "STUDI TENTANG PEMODELAN ONTOLOGI WEB SEMANTIK DAN PROSPEK PENERAPAN PADA BIBLIOGRAFI ARTIKEL JURNAL ILMIAH," in *Prosiding SAKTI (Seminar Ilmu Komputer dan Teknologi Informasi)*, 2017, vol. 2, no. 1, pp. 164–169.
- [3] A. Rochiyat and A. Wibowo, "Analisis Aktor Berpengaruh Dan Aktor Populer Dengan Metode Degree Centrality Dan Follower Rank Pada Tagar Twitter '# gejayanmemanggil,'" *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 6, no. 2, pp. 130–138, 2020.
- [4] M. S. Setatama and D. Tricahyono, "Implementasi Social Network Analysis pada Penyebaran Country Branding 'Wonderful Indonesia,'" *Indones. J. Comput.*, vol. 2, no. 2, pp. 91–104, 2017.
- [5] J. B. Sanawi, M. C. Samani, and M. Taibi, "# vaccination: Identifying influencers in the vaccination discussion on twitter through social network visualisation," *Int. J. Bus. Soc.*, vol. 18, no. S4, pp. 718–726, 2017.
- [6] I. Gialampoukidis, G. Kalpakis, T. Tsirikia, S. Papadopoulos, S. Vrochidis, and I. Kompatsiaris, "Detection of terrorism-related twitter communities using centrality scores," in *Proceedings of the 2nd International Workshop on Multimedia Forensics and Security*, 2017, pp. 21–25.
- [7] R. Oktora and A. Alamsyah, "Pola Interaksi dan Aktor yang Paling Berperan pada Event JGTC 2013 melalui Media Sosial Twitter (Studi Menggunakan Metode Social Network Analysis)," *J. Manaj. Indones.*, vol. 14, no. 3, pp. 201–209, 2014.
- [8] The W3C SPARQL Working Group, "SPARQL 1.1 Overview. W3C Recommendation 21 March 2013," W3C, 2013. .
- [9] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.
- [10] M. Needham and A. E. Hodler, *Graph Algorithms: Practical Examples in Apache Spark and Neo4j*. O'Reilly Media, 2019.
- [11] R. kumar Kaliyar, "Graph databases: A survey," in *International Conference on Computing, Communication & Automation*, 2015, pp. 785–790.
- [12] A. Hagberg, P. Swart, and D. S Chult, "Exploring network structure, dynamics, and function using NetworkX," Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.
- [13] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. Nov, pp. 2579–2605, 2008.
- [14] N. Tatti, "Density-friendly graph decomposition," *ACM Trans. Knowl. Discov. from Data*, vol. 13, no. 5, pp. 1–29, 2019.
- [15] "Daftar Film Indonesia." http://id.dbpedia.org/page/Daftar_film_Indonesia.
- [16] novyantsp, "novyantsp/notebook-tweb: Notebook Tweb," May 2020, doi: 10.5281/ZENODO.3839044.

