

TUGAS AKHIR - KM184801

**ANALISIS JARINGAN SOSIAL PERCAKAPAN
PENGGUNA TWITTER MENGENAI PERUSAHAAN
EDUCATIONAL TECHNOLOGY MENGGUNAKAN
METRIK *NETWORK PROPERTIES* DAN *CENTRALITY***

ADRIAN MAULANA MUHAMMAD

NRP 06111540000099

Dosen Pembimbing

Dr. Imam Mukhlash, S.Si, MT

NIP 19700831 199403 1 003

Dr. Darmaji, S.Si, MT

NIP 19691015 199412 1 001

Program Studi S-1

Departemen Matematika

Fakultas Sains dan Analitika Data

Institut Teknologi Sepuluh Nopember

Surabaya

2022



TUGAS AKHIR - KM184801

**ANALISIS JARINGAN SOSIAL PERCAKAPAN
PENGGUNA TWITTER MENGENAI PERUSAHAAN
EDUCATIONAL TECHNOLOGY MENGGUNAKAN
METRIK *NETWORK PROPERTIES* DAN *CENTRALITY***

ADRIAN MAULANA MUHAMMAD

NRP 06111540000099

Dosen Pembimbing

Dr. Imam Mukhlash, S.Si, MT

NIP 19700831 199403 1 003

Dr. Darmaji, S.Si, MT

NIP 19691015 199412 1 001

Program Studi S-1

Departemen Matematika

Fakultas Sains dan Analitika Data

Institut Teknologi Sepuluh Nopember

Surabaya

2022



FINAL PROJECT - KM184801

**SOCIAL NETWORK ANALYSIS OF TWITTER USER
CONVERSATIONS ABOUT EDUCATIONAL
TECHNOLOGY COMPANIES USING NETWORK
PROPERTIES AND CENTRALITY METRICS**

ADRIAN MAULANA MUHAMMAD

NRP 06111540000099

Advisors

Dr. Imam Mukhlash, S.Si, MT

NIP 19700831 199403 1 003

Dr. Darmaji, S.Si, MT

NIP 19691015 199412 1 001

Study Program Bachelor

Department of Mathematics

Faculty of Science and Data Analytics

Institut Teknologi Sepuluh Nopember

Surabaya

2022

LEMBAR PENGESAHAN

ANALISIS JARINGAN SOSIAL PERCAKAPAN PENGGUNA TWITTER MENGENAI PERUSAHAAN *EDUCATIONAL TECHNOLOGY* MENGGUNAKAN METRIK *NETWORK PROPERTIES* DAN *CENTRALITY*

TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat
memperoleh gelar Sarjana Matematika pada

Program Studi S-1

Departemen Matematika

Fakultas Sains dan Analitika Data

Institut Teknologi Sepuluh Nopember

Oleh: **ADRIAN MAULANA MUHAMMAD**

NRP. 06111540000099

Disetujui oleh Tim Penguji Tugas Akhir:

- | | |
|-------------------------------------|---------------|
| 1. Dr. Imam Mukhlash, S.Si, MT | Pembimbing I |
| 2. Dr. Darmaji, S.Si, MT | Pembimbing II |
| 3. Alvida Mustika Rukmi, S.Si, M.Si | Penguji |
| 4. Dr. Budi Setiyono, S.Si, MT | Penguji |
| 5. Drs. Suhud Wahyudi, M.Si | Penguji |

SURABAYA

Juli, 2022

PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini:

Nama mahasiswa / NRP : Adrian Maulana Muhammad / 06111540000099
Program studi : S-1 Matematika
Dosen Pembimbing / NIP : Dr. Imam Mukhlash, S.Si, MT / 19700831 199403 1 003
Dr. Darmaji, S.Si, MT / 19691015 199412 1 001

dengan ini menyatakan bahwa Tugas Akhir dengan judul “Analisis Jaringan Sosial Percakapan Pengguna Twitter Mengenai Perusahaan *Educational Technology* Menggunakan Metrik *Network Properties* dan *Centrality*” adalah hasil karya sendiri, bersifat orisinal, dan ditulis dengan mengikuti kaidah penulisan ilmiah.

Bilamana di kemudian hari ditemukan ketidaksesuaian dengan pernyataan ini, maka saya bersedia menerima sanksi sesuai dengan ketentuan yang berlaku di Institut Teknologi Sepuluh Nopember.

Surabaya, 28 Juli 2022

Mengetahui,

Dosen Pembimbing I

Dosen Pembimbing II

Dr. Imam Mukhlash, S.Si, MT
NIP. 19700831 199403 1 003

Dr. Darmaji, S.Si, MT
NIP. 19691015 199412 1 001

Mahasiswa

Adrian Maulana Muhammad
NRP. 06111540000099

ABSTRAK

ANALISIS JARINGAN SOSIAL PERCAKAPAN PENGGUNA TWITTER MENGENAI PERUSAHAAN *EDUCATIONAL TECHNOLOGY* MENGGUNAKAN METRIK *NETWORK PROPERTIES* DAN *CENTRALITY*

Nama Mahasiswa / NRP : Adrian Maulana Muhammad / 06111540000099
Departemen : Matematika FSAD - ITS
Dosen Pembimbing : 1. Dr. Imam Mukhlash, S.Si, MT
2. Dr. Darmaji, S.Si, MT

Abstrak

Jumlah pengguna media sosial di Indonesia meningkat 10 juta pengguna atau sebesar 6,3 persen di antara tahun 2020 dan 2021. Fenomena ini dapat membuka peluang baru bagi perusahaan untuk meningkatkan efektivitas kegiatan pemasarannya di media sosial, termasuk bagi perusahaan di sektor *educational technology* (*edtech*). Ruangguru dan Zenius merupakan perusahaan *edtech* yang paling dominan di dalam persaingan industri *edtech* di Indonesia, baik dari pandangan investor maupun dalam jumlah *followers* di Twitter. Twitter dapat dimanfaatkan oleh perusahaan sebagai salah satu media pemasaran agar dapat menjangkau audiens yang lebih luas dibandingkan dengan perusahaan pesaing. Oleh karena itu, perusahaan perlu mengetahui bagaimana aktivitas pemasaran atau *brand recognition* perusahaan tersebut di Twitter dibandingkan dengan perusahaan pesaing. Penelitian ini menggunakan penerapan *Social Network Analysis* (SNA) dalam menganalisis jaringan sosial percakapan pengguna Twitter mengenai kedua perusahaan *edtech* tersebut. Metrik SNA yang digunakan pada penelitian ini adalah metrik *network properties* dan metrik *centrality*. Berdasarkan hasil yang didapatkan pada perhitungan metrik *network properties*, struktur jaringan sosial percakapan pengguna Twitter mengenai Zenius lebih unggul dibandingkan Ruangguru, dengan unggul lima dari delapan metrik, dengan pengguna yang paling berpengaruh terhadap alur penyebaran informasi di masing-masing jaringan, didominasi oleh akun Schfess, berdasarkan hasil perhitungan metrik *centrality*. Hal ini menunjukkan bahwa pada jaringan Zenius, persebaran informasi lebih efisien dibandingkan jaringan Ruangguru.

Kata kunci: *Ruangguru, Social Network Analysis, Twitter, Zenius.*

ABSTRACT

SOCIAL NETWORK ANALYSIS OF TWITTER USER CONVERSATIONS ABOUT EDUCATIONAL TECHNOLOGY COMPANIES USING NETWORK PROPERTIES AND CENTRALITY METRICS

Student Name / NRP : Adrian Maulana Muhammad / 06111540000099
Department : Mathematics FSDA - ITS
Advisors : 1. Dr. Imam Mukhlash, S.Si, MT
2. Dr. Darmaji, S.Si, MT

Abstract

The number of social media users in Indonesia has increased by 10 million users or 6.3 percent between 2020 and 2021. This phenomenon could open up new opportunities for companies to increase the effectiveness of their marketing activities on social media, including for educational technology (edtech) companies. Ruangguru and Zenius are the most dominant edtech companies in Indonesia, both from the perspective of investors and in terms of the number of followers on Twitter. Twitter can be used by companies as a marketing medium in order to reach a wider audience than the competing companies. Therefore, companies need to know how the company's marketing activities or brand recognition on Twitter compared to the competing companies. This study uses the application of Social Network Analysis (SNA) in analyzing the interaction of social media users between the two edtech companies. The SNA metrics used in this study are network properties and centrality metrics. Based on the results obtained in the calculation of the network properties metrics, the social network structure of Twitter users' conversations about Zenius is superior to Ruangguru, by winning five network properties' metrics, with a user who has the most influence on the flow of information dissemination dominated by the Schfess account. This shows that on the Zenius network, the distribution of information is more efficient than on the Ruangguru network.

Keywords: *Ruangguru, Social Network Analysis, Twitter, Zenius.*

KATA PENGANTAR

Alhamdulillahirabbil'alamin, puji syukur saya panjatkan kepada Allah SWT yang telah memberikan limpahan rahmat, petunjuk, serta hidayah, sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul "Analisis Jaringan Sosial Percakapan Pengguna Twitter Mengenai Perusahaan *Educational Technology* Menggunakan Metrik *Network Properties* dan *Centrality*" sebagai salah satu syarat kelulusan Program Sarjana Departemen Matematika FSAD Institut Teknologi Sepuluh Nopember (ITS) Surabaya. Penulis ingin menyampaikan terima kasih kepada semua pihak yang telah membantu dan mendukung penulis hingga terselesainya Tugas Akhir ini:

1. Pertama saya ucapkan terima kasih kepada orang tua penulis serta saudara kandung penulis yang selalu mendoakan dan mendukung penulis selama menempuh perkuliahan di Departemen Matematika ITS.
2. Bapak Dr. Imam Mukhlash, S.Si, MT dan Bapak Dr. Darmaji, S.Si, MT selaku dosen pembimbing atas segala arahan, dukungan, dan motivasinya kepada penulis, sehingga Tugas Akhir ini dapat terselesaikan dengan baik.
3. Bapak Budi Setiyono, S.Si, MT, Bapak Drs. Suhud Wahyudi, M.Si, dan Ibu Alvida Mustika Rukmi, S.Si, M.Si selaku dosen penguji yang telah memberikan masukan dan saran yang membangun dalam menyelesaikan Tugas Akhir.
4. Bapak Subchan, Ph.D selaku Kepala Departemen Matematika Institut Teknologi Sepuluh Nopember yang telah memberikan arahan akademis selama penulis kuliah di Departemen Matematika ITS.
5. Ibu Dr. Dwi Ratna Sulistyningrum, MT. dan Bapak Dr. Budi Setiyono, S.Si, MT. selaku Sekretaris Departemen Matematika ITS yang telah memberikan arahan akademis selama penulis kuliah di Departemen Matematika ITS.
6. Bapak Prof. Dr. Drs. Subiono, M.Sc. selaku dosen wali yang telah memberikan nasihat dan arahan selama penulis menempuh perkuliahan di Departemen Matematika ITS.
7. Seluruh Bapak dan Ibu Dosen dan Staff Departemen Matematika ITS yang telah memberikan ilmu, bimbingan, dan motivasi kepada penulis selama kuliah di Departemen Matematika ITS.
8. Semua pihak yang tidak bisa ditulis satu persatu, terima kasih sudah memberikan motivasi dan dukungan kepada penulis.

Penulis menyadari bahwa dalam Tugas Akhir ini masih terdapat kekurangan. Oleh karena itu, kritik dan saran yang bersifat membangun sangat diharapkan oleh penulis. Semoga Tugas Akhir ini dapat bermanfaat bagi penulis dan semua pihak yang berkepentingan.

Surabaya, 25 Juli 2022

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN	vii
PERNYATAAN ORISINALITAS	ix
ABSTRAK	xi
ABSTRACT.....	xiii
KATA PENGANTAR.....	xv
DAFTAR ISI.....	xvii
DAFTAR GAMBAR.....	xix
DAFTAR TABEL	xxi
DAFTAR KODE PROGRAM.....	xxiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan.....	3
1.5 Manfaat.....	3
BAB II TINJAUAN PUSTAKA	5
2.1 Penelitian Terdahulu.....	5
2.2 Dasar Teori	6
BAB III METODOLOGI	25
3.1. Objek dan Aspek Penelitian	25
3.2. Peralatan Penunjang Penelitian	25
3.3. Tahapan Pelaksanaan Penelitian.....	26
BAB IV HASIL DAN PEMBAHASAN.....	29
4.1. <i>Data Collection</i>	29
4.2. <i>Data Pre-processing</i>	29
4.3. <i>Social Network Analysis (SNA)</i>	32
4.4. Visualisasi Model Jaringan.....	43
4.5. Hasil SNA dan Performa Perusahaan Ruangguru dan Zenius	45
BAB V KESIMPULAN DAN SARAN.....	47
5.1. Kesimpulan.....	47
5.2. Saran	47
DAFTAR PUSTAKA.....	48
LAMPIRAN.....	49
LAMPIRAN.....	50
BIODATA PENULIS.....	57

DAFTAR GAMBAR

Gambar 1.1 Pertumbuhan Perusahaan <i>Edtech</i> di Indonesia	1
Gambar 1.2 Tingkat Penetrasi Internet di Indonesia.....	Error! Bookmark not defined.
Gambar 1.3 Jumlah Percakapan Pengguna Twitter Mengenai Ruangguru & Zenius	2
Gambar 2.1 Representasi Graf dengan <i>Adjacency Matrix</i> dan <i>Edge List</i>	11
Gambar 2.2 Contoh Ilustrasi Perhitungan Metrik <i>Density</i>	15
Gambar 2.3 Ilustrasi Algoritma Louvain	16
Gambar 2.4 <i>Pseudocode</i> Algoritma Louvain.....	17
Gambar 2.5 Contoh Implementasi Algoritma BFS pada Jaringan	18
Gambar 2.6 Contoh Ilustrasi Perhitungan Metrik <i>Average Path Length</i>	18
Gambar 2.7 Contoh Ilustrasi Perhitungan Metrik <i>Average Degree</i>	19
Gambar 2.8 Contoh Ilustrasi dan <i>Pseudocode</i> Algoritma <i>Connected Components</i>	20
Gambar 2.9 Dua Aktor dengan Nilai <i>Degree Centrality</i> yang Sama.....	21
Gambar 2.10 Contoh Ilustrasi Perhitungan Metrik <i>Degree Centrality</i>	21
Gambar 2.11 Contoh Ilustrasi Perhitungan Metrik <i>Betweenness Centrality</i>	22
Gambar 2.12 Contoh Ilustrasi Perhitungan Metrik <i>Closeness Centrality</i>	23
Gambar 3.1 Diagram Alir Penelitian	Error! Bookmark not defined.
Gambar 3.2 Diagram Alir Penelitian (lanjutan).....	Error! Bookmark not defined.
Gambar 4.1 Visualisasi Jaringan Ruangguru	41
Gambar 4.2 Visualisasi Jaringan Zenius	41

DAFTAR TABEL

Tabel 2.1	Sepuluh Perusahaan <i>Edtech</i> Indonesia yang Mendapat Perhatian Investor	6
Tabel 2.2	Terminologi pada Teori Graf dan SNA	11
Tabel 2.3	Pengukuran dan Metrik pada SNA	12
Tabel 2.4	Rangkuman Metrik <i>Network Properties</i>	20
Tabel 2.5	Rangkuman Metrik <i>Centrality</i>	24
Tabel 3.1	Spesifikasi Perangkat	23
Tabel 4.1	Nilai <i>Size</i> dan <i>Order</i> pada Kedua Jaringan	31
Tabel 4.2	Nilai <i>Density</i> pada Kedua Jaringan	31
Tabel 4.3	Nilai <i>Modularity</i> pada Kedua Jaringan	32
Tabel 4.4	Nilai <i>Diameter</i> pada Kedua Jaringan	33
Tabel 4.5	Nilai <i>Average Path Length</i> pada Kedua Jaringan	33
Tabel 4.6	Nilai <i>Average Degree</i> pada Kedua Jaringan	34
Tabel 4.7	Nilai <i>Connected Components</i> pada Kedua Jaringan	34
Tabel 4.8	Perbandingan Metrik <i>Network Properties</i> pada Jaringan Ruangguru dan Zenius	35
Tabel 4.9	Lima Aktor dengan <i>Degree Centrality</i> Tertinggi pada Jaringan Ruangguru	36
Tabel 4.10	Lima Aktor dengan <i>Degree Centrality</i> Tertinggi pada Jaringan Zenius	36
Tabel 4.11	Lima Aktor dengan <i>Betweenness Centrality</i> Tertinggi pada Jaringan Ruangguru	37
Tabel 4.12	Lima Aktor dengan <i>Betweenness Centrality</i> Tertinggi pada Jaringan Zenius	37
Tabel 4.13	Lima Aktor dengan <i>Closeness Centrality</i> Tertinggi pada Jaringan Ruangguru	38
Tabel 4.14	Lima Aktor dengan <i>Closeness Centrality</i> Tertinggi pada Jaringan Zenius	38
Tabel 4.15	Lima Aktor dengan <i>Eigenvector Centrality</i> Tertinggi pada Jaringan Ruangguru	39
Tabel 4.16	Lima Aktor dengan <i>Eigenvector Centrality</i> Tertinggi pada Jaringan Zenius	39
Tabel 4.17	Seluruh Hasil Perhitungan Metrik <i>Centrality</i> pada Jaringan Ruangguru	40
Tabel 4.18	Seluruh Hasil Perhitungan Metrik <i>Centrality</i> pada Jaringan Zenius	40
Tabel 4.19	Lima Kelompok Terbesar pada Jaringan Ruangguru	41
Tabel 4.20	Lima Kelompok Terbesar pada Jaringan Zenius	42

DAFTAR KODE PROGRAM

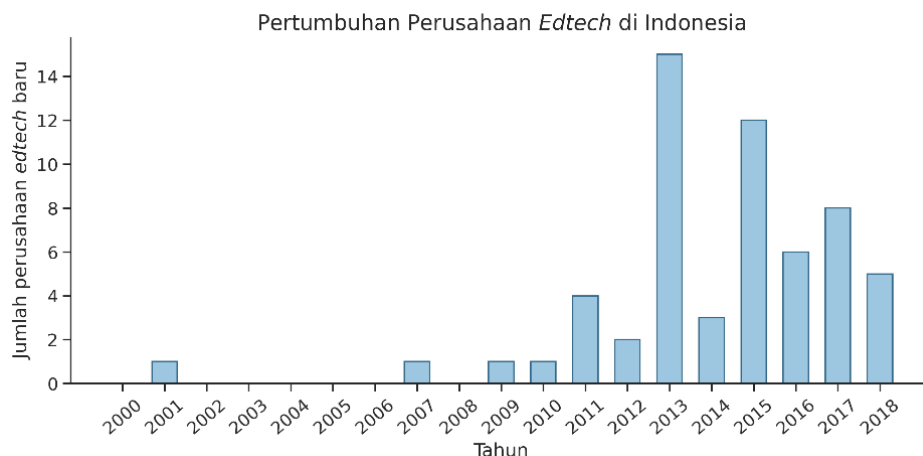
Kode Program 4.1 <i>Scraping Data Twitter</i>	27
Kode Program 4.2 <i>Import Dataset</i>	30
Kode Program 4.3 Menghapus <i>Tweet</i> yang Duplikat	30
Kode Program 4.4 Menghapus <i>Tweet</i> yang Tidak Mempunyai Interaksi	30
Kode Program 4.5 Mengambil <i>Tweet</i> Percakapan Antara Pengguna	31
Kode Program 4.6 Pengelompokan <i>Tweet</i> Mengenai Kedua Perusahaan <i>Edtech</i>	31
Kode Program 4.7 Transformasi Data ke Bentuk <i>Edge List</i>	32
Kode Program 4.8 Transformasi <i>Edge List</i> ke Graf Tanpa Arah	32
Kode Program 4.9 <i>Order</i> dan <i>Size</i> pada Jaringan	33
Kode Program 4.10 <i>Density</i> pada Jaringan	33
Kode Program 4.11 <i>Modularity</i> pada Jaringan	32
Kode Program 4.12 <i>Diameter</i> pada Jaringan	32
Kode Program 4.13 <i>Average Path Length</i> pada Jaringan	33
Kode Program 4.14 <i>Average Degree</i> pada Jaringan	34
Kode Program 4.15 <i>Connected Components</i> pada Jaringan	34
Kode Program 4.16 <i>Degree Centrality</i> pada Jaringan	36
Kode Program 4.17 <i>Betweenness Centrality</i> pada Jaringan	37
Kode Program 4.18 <i>Closeness Centrality</i> pada Jaringan	40
Kode Program 4.19 <i>Eigenvector Centrality</i> pada Jaringan	41
Kode Program 4.20 Transformasi Data Graf Networkx ke <i>File GEXF</i>	40

BAB I

PENDAHULUAN

1.1 Latar Belakang

Berdasarkan laporan dari Data Reportal pada tahun 2021, Indonesia merupakan salah satu negara dengan tingkat penggunaan internet yang tinggi, dengan jumlah pengguna internet meningkat 27 juta pengguna atau sebesar 16 persen dan jumlah pengguna media sosial meningkat 10 juta pengguna atau sebesar 6,3 persen di antara tahun 2020 dan 2021 (Data Reportal, 2021). Penggunaan internet yang meningkat di Indonesia, membuka peluang munculnya suatu inovasi pada penggunaan informasi, komunikasi, dan teknologi dalam bidang pendidikan, yaitu platform *educational technology* (*edtech*). Produk *edtech* di Indonesia secara umum bertujuan untuk membantu murid dalam belajar dan meningkatkan keterampilan mereka dengan konten *e-learning* yang dapat membantu murid dalam memahami tugas atau persiapan ujian. Secara keseluruhan, sektor *edtech* di Indonesia masih dalam fase pertumbuhan, dengan hampir seluruh perusahaan masih terus melakukan eksperimen pada produk ataupun pasar mereka, meskipun demikian, terdapat persepsi umum pada investor bahwa sektor ini memiliki potensi pasar yang besar. Sebagian besar perusahaan *edtech* di Indonesia didirikan dalam jangka waktu enam tahun terakhir dengan rata-rata jumlah tahun operasi sekitar empat tahun. Pada Gambar 1.1 menunjukkan pertumbuhan pesat perusahaan *edtech* dalam beberapa tahun terakhir di Indonesia (Riaz dkk., 2020).

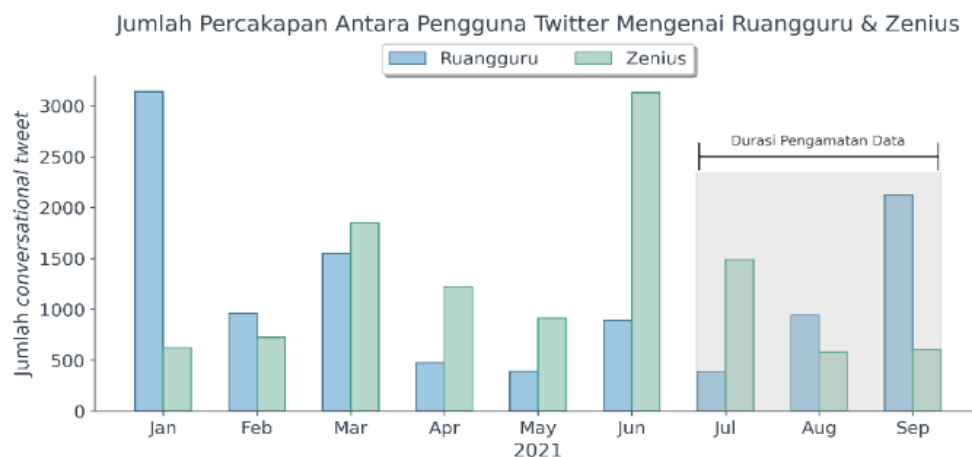


Gambar 1.1 Pertumbuhan Perusahaan *Edtech* di Indonesia
Sumber Data: (Riaz dkk., 2020)

Dua perusahaan *edtech* di Indonesia, yaitu Ruangguru dan Zenius, termasuk dalam daftar perusahaan *edtech* yang paling dominan dalam hal pertumbuhan pengguna dan pendanaan investor selama beberapa tahun terakhir di Indonesia. Selain itu, kedua perusahaan tersebut juga merupakan akun platform *edtech* di Indonesia yang memiliki jumlah pengikut atau *followers* terbanyak pada situs jejaring sosial *online*, Twitter (Tracxn, 2022). Namun, dengan memiliki jumlah *followers* Twitter yang besar, tidak berarti banyak jika tingkat interaksi rendah dan perusahaan tidak melibatkan *followers* tersebut dalam kegiatan pemasaran. Pengguna Twitter atau *followers* tersebut tidak hanya sebagai penerima informasi, tetapi juga dapat berpartisipasi aktif dalam pengembangan dan penyebaran *brand recognition* dari perusahaan tersebut. Oleh karena itu, diperlukan suatu analisis yang dapat membantu perusahaan dalam memahami pola interaksi pengguna Twitter, sehingga perusahaan dapat mengidentifikasi bagaimana aktivitas pemasaran produk mereka di Twitter dibandingkan dengan perusahaan pesaing.

Salah satu metode dalam *social media analytics* yang umum digunakan untuk menganalisis pola interaksi antara individu atau kelompok adalah *Social Network Analysis* (SNA). SNA merupakan suatu pendekatan analitis yang memanfaatkan teori graf untuk memodelkan struktur suatu jaringan sosial. Suatu jaringan sosial dapat terdiri dari pengguna Twitter, dilambangkan dengan simpul, dan interaksi percakapan antara pengguna Twitter tersebut, dilambangkan dengan sisi. Dengan menggunakan pendekatan SNA, perusahaan dapat menganalisis struktur jaringan sosial yang terbentuk berdasarkan interaksi percakapan pengguna Twitter (*word-of-mouth*) mengenai perusahaan tersebut, dan dengan pengetahuan ini, perusahaan dapat mengetahui bagaimana aktivitas pemasaran atau *brand recognition* perusahaan tersebut di Twitter dibandingkan dengan perusahaan pesaing.

Pada Gambar 1.3 terlihat bahwa, sekitar awal Juli 2021 sampai dengan akhir September 2021, jumlah percakapan pengguna Twitter mengenai Zenius mengalami penurunan, sedangkan jumlah percakapan mengenai Ruangguru terus meningkat, bahkan melebihi jumlah percakapan mengenai Zenius. Durasi waktu ini menarik, karena dapat terlihat apakah pergerakan tren jumlah percakapan pengguna Twitter mengenai kedua perusahaan tersebut berbanding lurus atau tidak dengan kualitas struktur jaringan sosial yang terbentuk. Pada SNA, struktur suatu jaringan sosial dapat dianalisis dengan metrik *network properties*. Selain itu, hal yang menarik untuk dianalisis adalah *key actor* atau pengguna Twitter yang mempunyai pengaruh besar di dalam jaringan sosial percakapan tersebut. Pada SNA, *key actor* di dalam suatu jaringan sosial dapat diidentifikasi dengan metrik *centrality*. Durasi pengamatan data ditunjukkan pada Gambar 1.3.



Gambar 1.2 Jumlah Percakapan Pengguna Twitter Mengenai Ruangguru & Zenius
Sumber Data: (Twitter, 2021)

Terdapat beberapa penelitian terdahulu yang membahas penerapan SNA dalam perumusan strategi pemasaran di media sosial. Pertama, penelitian yang dilakukan oleh Ioannis Antoniadis dan Anna Charmantzi tentang penerapan SNA dalam membangun strategi komunikasi dan *branding* dengan membangun *social capital* di situs jejaring sosial (Antoniadis & Charmantzi, 2016). Kedua, penelitian yang dilakukan oleh Arnaldo Litterio dkk. mengenai penerapan SNA dalam pemasaran untuk mengidentifikasi *opinion leaders* (Litterio dkk., 2017). Ketiga, penelitian yang dilakukan oleh Itai Himelboim dan Guy Golan mengenai pendekatan *social networks* dalam menganalisis peran *influencer* pada *viral advertising* (Himelboim & Golan, 2019). Berdasarkan latar belakang yang telah diuraikan di atas, maka penulis melakukan suatu penelitian tentang analisis jaringan sosial percakapan pengguna Twitter mengenai perusahaan *edtech*, yaitu Ruangguru dan Zenius, menggunakan metrik *network properties* dan *centrality*.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, rumusan masalah pada penelitian ini sebagai berikut:

1. Bagaimana penerapan SNA dalam menganalisis dan membandingkan struktur jaringan sosial percakapan pengguna Twitter mengenai Ruangguru dan Zenius?
2. Bagaimana penerapan SNA dalam mengidentifikasi pengguna yang paling berpengaruh (*key actor*) di dalam jaringan sosial percakapan pengguna Twitter mengenai Ruangguru dan Zenius?

1.3 Batasan Masalah

Penulis membatasi permasalahan pada penelitian ini sebagai berikut:

1. Data yang digunakan pada penelitian ini adalah kumpulan data *tweets* sebesar 39.219 baris dan 6 kolom yang diperoleh dari Twitter dengan kata kunci “ruangguru” (terdiri dari 5.488 pengguna dan 4.982 percakapan) dan “zenius” (terdiri dari 2.605 pengguna dan 2.123 percakapan) sejak 1 Juli 2021 sampai dengan 30 September 2021.
2. Jenis interaksi antara pengguna Twitter yang digunakan pada penelitian ini adalah interaksi *reply* atau interaksi yang berbentuk percakapan atau tanggapan dari pengguna Twitter atas suatu *tweet*.
3. Jenis graf yang digunakan pada SNA di penelitian ini adalah berupa graf tanpa arah (*undirected graph*).
4. Pengukuran SNA pada penelitian ini dibagi menjadi dua bagian, yaitu *network properties* dan *centrality*, dengan beberapa metrik pada setiap pengukurannya.
5. Proses pengumpulan data pada penelitian ini dilakukan dengan metode *scraping* pada situs jejaring *online* Twitter yang dapat diakses bebas oleh publik.

1.4 Tujuan

Tujuan dalam penelitian ini adalah sebagai berikut:

1. Menganalisis dan membandingkan struktur jaringan sosial percakapan pengguna Twitter mengenai Ruangguru dan Zenius dengan menggunakan metrik *network properties*.
2. Mengidentifikasi pengguna Twitter yang paling berpengaruh (*key actor*) di dalam jaringan percakapan mengenai Ruangguru dan Zenius di Twitter dengan menggunakan metrik *centrality*.

1.5 Manfaat

Manfaat yang diharapkan dari penelitian ini adalah sebagai berikut:

1. Penelitian ini dapat bermanfaat sebagai bahan pertimbangan atau referensi kepada kedua perusahaan di sektor *edtech*, yaitu Ruangguru dan Zenius, dalam hal strategi pemasaran di media sosial Twitter.
2. Dalam bidang akademik, penelitian ini mempunyai manfaat untuk mengetahui dan memahami pendekatan SNA dalam menganalisis struktur suatu jaringan sosial yang terbentuk berdasarkan interaksi pengguna media sosial.

BAB II TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Terdapat tiga penelitian terdahulu yang menjadi referensi utama pada penelitian ini. Pertama, penelitian yang ditulis oleh Ioannis Antoniadis dan Anna Charmantzi tentang penerapan SNA dalam membangun strategi komunikasi dan *branding* dengan membangun *social capital* di situs jejaring sosial *online*. Penelitian ini bertujuan untuk menunjukkan pentingnya SNA dan *social capital* sebagai instrumen untuk pengambilan keputusan dalam strategi pemasaran di media sosial. Studi kasus dalam penelitian ini adalah suatu halaman Facebook merek produk tradisional dari Yunani yang sedang merubah strategi pemasarannya untuk menarik konsumen dari kalangan muda. Data dikumpulkan dengan perangkat lunak NodeXL. Metrik pada SNA yang digunakan pada penelitian ini adalah *network properties: size, order, density, average path length, connected components*; dan metrik *centrality: betweenness centrality, closeness centrality, eigenvector centrality*. Temuan pada penelitian ini adalah kegiatan pemasaran dengan tema kontes atau permainan terbukti menjadi strategi yang baik, karena tersebar luas dan banyak berinteraksi dengan para pengguna. Kesimpulan pada penelitian ini adalah mengenai bagian yang penting dari kesuksesan aktivitas pemasaran di media sosial adalah pemahaman tentang struktur jaringan sosial dan *social capital* yang dibuat di dalamnya. Menganalisis dan memahami mekanisme ini dengan SNA dapat memberikan wawasan yang berguna, baik bagi praktisi maupun akademisi, untuk memandu keputusan terkait manajemen pemasaran, media sosial, dan pemasaran konten (Antoniadis & Charmantzi, 2016).

Kedua, penelitian yang ditulis oleh Arnaldo Litterio dkk. mengenai penerapan SNA dalam pemasaran untuk mengidentifikasi *opinion leaders*. Tujuan dari penelitian ini adalah menerapkan pendekatan SNA untuk mengidentifikasi *influencer* yang potensial atau aktor yang signifikan dari komunitas *online* dalam sudut pandang pemasaran. Metode yang digunakan untuk mengklasifikasikan individu di dalam jaringan sosial yang terbentuk adalah dengan kombinasi metrik *eigenvector centrality* dan *betweenness centrality*. Studi kasus pada penelitian ini adalah pada suatu situs halaman Facebook mengenai acara olahraga. Perangkat lunak NodeXL digunakan untuk mengekstrak dan menganalisis informasi dari situs tersebut. Temuan pada penelitian ini adalah model yang diusulkan efektif dalam mendeteksi aktor yang berpotensi menyebarkan pesan secara efisien dengan komponen lainnya dari komunitas, hal tersebut dicapai berdasarkan posisi aktor tersebut di dalam jaringan. Kesimpulan pada penelitian ini adalah SNA dapat berguna untuk mendeteksi *subgroups* dengan karakteristik tertentu yang mungkin tidak terlihat dari metode analisis lainnya (Litterio dkk., 2017).

Ketiga, penelitian yang ditulis oleh Itai Himelboim dan Guy Golan mengenai pendekatan *social networks* dalam menganalisis peran *influencer* pada *viral advertising*. Penelitian ini mengidentifikasi tiga kelompok *influencer* berdasarkan konektivitas mereka di jaringan. Pengguna yang paling banyak di *retweet* adalah *primary influencers*, pengguna yang paling banyak di *mention* adalah *contextual influencers*, dan pengguna lainnya yang memiliki *low influence*. Studi kasus pada penelitian ini adalah *marketing campaign* Heineken. Data dikumpulkan dari semua pengguna Twitter yang *posting tweet* berisi tautan ke iklan Heineken di YouTube, beserta *retweet* dan *mention* dari *tweet* tersebut. Metode yang digunakan pada penelitian ini adalah metrik *degree centrality* dan *betweenness centrality*. Kesimpulan pada penelitian ini adalah analisis jaringan adalah satu-satunya metode yang memungkinkan representasi yang berarti dari proses distribusi *viral advertising* (Himelboim & Golan, 2019).

2.2 Dasar Teori

Pada bagian ini akan dijelaskan mengenai teori-teori yang digunakan sebagai landasan dasar dalam pengerjaan penelitian ini. Beberapa hal yang akan dijelaskan pada bagian ini adalah mengenai *educational technology*, Ruangguru, Zenius, teori graf, *data mining*, dan SNA.

2.2.1 Educational Technology

Salah satu bentuk reformasi dalam sistem pendidikan publik adalah dengan penggunaan informasi, komunikasi, dan teknologi untuk pendidikan, yaitu platform *educational technology* (*edtech*) yang bertujuan untuk meningkatkan kualitas pendidikan publik. Hal ini terjadi terutama di Indonesia, di mana telah menunjukkan minat yang kuat untuk memanfaatkan teknologi untuk pembelajaran. Produk *edtech* di Indonesia secara umum bertujuan untuk membantu murid dalam belajar dan meningkatkan keterampilan mereka dengan konten *e-learning*, serta layanan *online* interaktif yang dapat membantu murid dalam memahami tugas atau persiapan ujian. Secara global, tren munculnya perusahaan *edtech* sebagian besar didorong oleh sektor swasta, yang memiliki insentif lebih tinggi untuk berinovasi daripada sektor publik. Sektor *edtech* di Indonesia mulai mengejar ketertinggalan global dengan munculnya tren pertumbuhan platform *edtech* dalam jangka waktu enam tahun terakhir dengan rata-rata tahun operasi adalah sekitar empat tahun. Pada tahun 2020, Indonesia telah memiliki 60 perusahaan atau *startup* pada sektor *edtech* yang aktif beroperasi. Hal ini menunjukkan banyaknya perusahaan pada industri *edtech* di Indonesia, sehingga dapat meningkatkan persaingan. Peningkatan dalam pendirian perusahaan *edtech* juga bersesuaian dengan tingkat penetrasi internet di Indonesia. Secara keseluruhan, sektor *edtech* di Indonesia masih dalam fase pertumbuhan, dengan hampir seluruh perusahaan masih terus melakukan eksperimen pada produk ataupun pasar mereka, meskipun demikian, terdapat persepsi umum pada investor bahwa sektor industri *edtech* di Indonesia memiliki potensi pasar yang besar (Riaz dkk., 2020).

Berdasarkan laporan oleh Tracxn, di mana merupakan platform terbesar di dunia untuk melacak entitas di seluruh sektor dan industri secara global, pada 10 Juni 2022, Indonesia telah memiliki 386 *startups* atau perusahaan *edtech* dengan sepuluh perusahaan yang paling banyak mendapat perhatian utama dari para investor yang ditunjukkan pada Tabel 2.1. Sepuluh perusahaan tersebut adalah Ruangguru, PT Zenius Education, CoLearn, Cakap, Pintek, HarukaEdu, Gredu, Hacktiv8, Circledoo, dan Pahamify (Tracxn, 2022).

Tabel 2.1 Sepuluh Perusahaan *Edtech* Indonesia yang Mendapat Perhatian Investor

No.	Perusahaan	Tahun Berdiri	Pendanaan (USD)	Jumlah <i>followers</i> Twitter
1.	Ruangguru	2014	205 M	163,500
2.	PT Zenius Education	2007	60 M	217,700
3.	CoLearn	2019	34 M	3.216
4.	Cakap	2015	13 M	1.323
5.	Pintek	2018	8 M	79
6.	HarukaEdu	2013	4 M	346
7.	Gredu	2016	4 M	2.953
8.	Haktiv8	2016	3 M	2.105
9.	Circledoo	2017	200K	536
10.	Pahamify	2017	150K	186.800

Sumber Data: (Tracxn, 2022)

Pada Tabel 2.1 menunjukkan hasil pemeringkatan *startup edtech* berdasarkan pendanaan dari investor, terlihat bahwa Ruangguru dan Zenius menduduki peringkat yang lebih tinggi dibandingkan *startups edtech* lain untuk peringkat seluruh kategori *startup* yang berasal dari Indonesia. Selain itu, kedua perusahaan ini juga memiliki jumlah pengikut atau *followers* yang besar pada situs jejaring *online*, Twitter. Twitter dapat dimanfaatkan oleh perusahaan sebagai salah satu media komunikasi pemasaran yang dapat memberikan jangkauan audiens yang lebih banyak dibandingkan perusahaan pesaing. Berdasarkan kedua hal tersebut, dapat dilihat bahwa perusahaan Ruangguru dan Zenius merupakan pesaing yang paling dominan dibandingkan perusahaan-perusahaan lain pada sektor *edtech* di Indonesia. Sehingga, persaingan kedua perusahaan tersebut dapat merepresentasikan persaingan di dalam industri *edtech* di Indonesia.

2.2.2 Ruangguru

Ruangguru merupakan perusahaan teknologi terbesar di Asia Tenggara yang berfokus pada layanan pendidikan. Ruangguru mengembangkan berbagai layanan belajar berbasis teknologi, termasuk layanan kelas virtual, platform ujian *online*, video belajar berlangganan, serta konten-konten pendidikan lainnya yang bisa diakses *web* dan aplikasi Ruangguru. Hingga Desember 2020, Ruangguru telah memiliki lebih dari 22 juta pengguna terdaftar dan memberikan akses kepada lebih dari 300.000 guru privat. Angka ini meningkat sebanyak 7 juta pengguna atau setara dengan 46 persen dibandingkan dengan data pada akhir Desember 2019. Tahun 2020 juga menjadi momentum penting bagi Ruangguru melalui kehadirannya di Thailand dengan *brand* StartDee. Di tahun pertamanya, aplikasi ini telah digunakan oleh lebih dari 200 ribu pengguna. Setahun sebelumnya, Ruangguru berhasil memasuki pasar Vietnam dengan *brand* Kien Guru. Kini jumlah pengguna Kien Guru lebih dari 700 ribu. Hal ini mengukuhkan Ruangguru sebagai salah satu perusahaan teknologi pendidikan terbesar di Asia Tenggara. Berbagai tantangan yang dihadapi dunia pendidikan selama pandemi mendorong Ruangguru meluncurkan berbagai inovasi produk dan layanan untuk membantu pelaksanaan Pembelajaran Jarak Jauh (PJJ), Ruangguru meluncurkan produk ruangkelas gratis, sistem tata kelola PJJ, yang telah dimanfaatkan oleh 13.000 sekolah di 440 kota/kabupaten di seluruh Indonesia (Ruangguru, 2021).

Selama kurun waktu Juli 2021 sampai dengan September 2021, akun Twitter resmi perusahaan Ruangguru telah melakukan beberapa *marketing campaign* sebagai usaha untuk meningkatkan interaksi pengguna Twitter dalam penyebaran informasi atau konten pemasaran perusahaan Ruangguru di Twitter, atau biasa disebut *brand recognition*. Ruangguru juga melibatkan dan menggunakan beberapa *brand ambassadors* beserta *hashtags* pada kegiatan tersebut. Kegiatan *marketing campaign* yang beberapa kali dilakukan oleh Ruangguru di Twitter ditunjukkan pada Tabel 2.2 sebagai berikut.

Tabel 2.2 Aktivitas *Marketing Campaign* pada Akun Twitter Ruangguru

No.	<i>Hashtags</i>	Total <i>Tweets</i>	Total <i>Replies</i>	<i>Brand Ambassador</i>
1.	#SelaluAdaHarapan	21	38	@amandamanopo @fikinaki__ @angga_yunandaaa @imanusman
2.	#Ruangguru	8	474	
3.	#BrandAmbassadorRuangguru	7	463	
4.	#RuangguruSquad	4	6	
5.	#TimRuangguru	3	3	

Sumber Data: (Twitter, 2021)

2.2.3 Zenius

Zenius adalah pelopor *startup* teknologi pendidikan di Indonesia yang menggunakan metode *adaptive learning*. Zenius memproduksi konten pendidikan untuk semua level pendidikan dari SD, SMP, dan SMA beserta persiapan ujian nasional dan tes masuk perguruan tinggi. Zenius memulai perjalanannya pada tahun 2008 dengan produk CD/DVD, sebelum beralih ke *website* pada tahun 2010. Hingga sekarang, Zenius berhasil meluncurkan *mobile apps* yang bisa diunduh melalui iOS dan Android. Zenius telah memiliki lebih dari 74,000 video pembelajaran dan puluhan ribu latihan soal. Platform ini juga diklaim memiliki konten pendidikan terbesar di Indonesia dengan lebih dari 90.000 video untuk siswa SD hingga SMA. Dengan peluncuran aplikasi Zenius, sebagian besar pelajar di Indonesia sudah bisa mengakses materi dan fitur belajar milik Zenius. Tercatat, pada tahun 2020 sampai 2021 kunjungan ke *website* zenius.net adalah sebanyak 38.167.048 dan Zenius memiliki lebih dari 20 juta pengguna. Pada tahun yang sama, Zenius berhasil meluncurkan fitur *personalized learning* melalui ZenCore. Diluncurkan sejak Juli 2020, ZenCore telah digunakan oleh 230.000 lebih pengguna. Selain itu, Zenius juga meluncurkan fitur berbasis kecerdasan buatan, yaitu ZenBot. Hingga kini, lebih dari 1,4 juta soal telah dikirimkan pengguna melalui fitur ini. Sementara untuk para guru, Zenius juga memiliki fitur untuk guru, yaitu ZenRu. Sejak diluncurkan pada November 2020, ZenRu mencatat lebih dari 250.000 guru sebagai pengguna (Zenius, 2021).

Selama kurun waktu Juli 2021 sampai dengan September 2021, akun Twitter resmi perusahaan Zenius telah melakukan beberapa *marketing campaign* sebagai usaha untuk meningkatkan interaksi pengguna Twitter dalam penyebaran informasi atau konten pemasaran perusahaan Zenius di Twitter, atau biasa disebut *brand recognition*. Zenius juga melibatkan dan menggunakan beberapa *brand ambassadors* beserta *hashtags* pada kegiatan tersebut. Kegiatan *marketing campaign* yang dilakukan oleh Zenius di Twitter ditunjukkan pada Tabel 2.3 sebagai berikut.

Tabel 2.3 Aktivitas *Marketing Campaign* pada Akun Twitter Zenius

No.	Hashtags	Total Tweets	Total Replies	Brand Ambassador
1.	#ZenUpdate	5	3	@insisani @rsembhada
2.	#Zweet17	2	7	
3.	#AduSeru	1	0	
4.	#ZenKlopedia	1	0	
5.	#MulaiDariManaAja	1	15	

Sumber Data: (Twitter, 2021)

2.2.4 Twitter

Twitter adalah situs *micro-blogging* yang memungkinkan pengguna untuk mengirim dan menerima pesan publik singkat yang disebut *tweets*. *Tweets* dibatasi tidak lebih dari 280 karakter, dan dapat menyertakan tautan ke blog, *website*, gambar, video, dan semua materi *online* lainnya. Pengguna dapat mengirimkan *tweet* kapan saja, dari komputer atau *smartphone*. Dengan mengikuti orang lain, pengguna dapat membentuk Twitter *feed* pribadi dan instan yang memenuhi berbagai minat pengguna. Terdapat beberapa terminologi dari Twitter, seperti *follow* yang merupakan istilah mengenai mengikuti pengguna-pengguna lain, dengan begitu semua *tweets* mereka akan muncul di *feed*. *Retweet* atau RT adalah istilah mengenai membagikan *tweet* pengguna lain yang telah dilihat di *feed*. *Reply* adalah istilah mengenai menanggapi *tweet* pengguna lain. *Mention* merupakan istilah mengenai ketika pengguna lain menyebut pengguna di dalam *tweet* (Mollett dkk., 2011).

Twitter adalah salah satu platform media sosial yang paling populer, menarik sekitar 255 juta pengguna aktif per bulan, dengan sekitar 500 juta *tweet* terkirim setiap hari. Audiens besar yang berpotensi dijangkau oleh Twitter menjadikannya alat yang sangat menarik bagi *brand* untuk berinteraksi dengan pelanggan mereka, sehingga Twitter dapat digunakan baik untuk pemasaran maupun periklanan. Kehadiran dan popularitas selebriti di Twitter juga menarik audiens ke media tersebut, dengan banyak *brand* sekarang menambahkan *tweet* selebriti sebagai strategi *endorsement* mereka, sehingga dapat meningkatkan konten *brand* mereka muncul di Twitter *feed* pengguna lain. Jadi, dapat disimpulkan bahwa Twitter memiliki manfaat saat ini untuk digunakan sebagai alat pemasaran, karena Twitter dapat memungkinkan *brand* untuk menjangkau kelompok konsumen yang lebih besar, dan menyebarkan konten *brand* lebih luas lagi (Soboleva dkk., 2015).

2.2.5 Data Mining

Teknik *data mining* dapat dimanfaatkan untuk *Social Network Analysis*, terutama untuk kumpulan data yang besar dan tidak dapat ditangani dengan metode tradisional. Teknik *data mining* dapat dimanfaatkan untuk mengekstrak informasi atau pengetahuan yang direpresentasikan di dalam struktur graf atau jaringan. Istilah *data mining* sering digunakan untuk merujuk pada seluruh proses penemuan pengetahuan atau biasa disebut *Knowledge Discovery from Data* (KDD). Teknik *data mining* adalah urutan iteratif dari langkah-langkah berikut.

1. *Data cleaning* (untuk menghilangkan *noise* dan data yang tidak konsisten)
2. *Data integration* (di mana beberapa sumber data dapat digabungkan)
3. *Data selection* (di mana data yang relevan dengan tujuan analisis diambil dari *database*)
4. *Data transformation* (di mana data diubah dan dikonsolidasikan ke dalam bentuk yang sesuai untuk *mining*)
5. *Data mining* (proses penting di mana metode *intelligent* diterapkan untuk mengekstrak pola data)
6. *Pattern evaluation* (untuk mengidentifikasi pola yang benar-benar menarik yang mewakili pengetahuan)
7. *Knowledge presentation* (di mana teknik visualisasi digunakan untuk menyajikan pengetahuan yang telah di-*mining* kepada pengguna)

Langkah 1 hingga 4 adalah berbagai bentuk dari *data pre-processing*, di mana mempersiapkan data sebelum proses *mining*. Pola yang menarik disajikan kepada pengguna dan dapat disimpan sebagai suatu *insights* atau pengetahuan. Dalam perspektif *data mining*, SNA juga disebut *link analysis* atau *link mining* (Han dkk., 2011).

2.2.6 Teori Graf

Social Network Analysis atau analisis jaringan sosial adalah proses menganalisis struktur jaringan sosial melalui pemanfaatan teori graf untuk memodelkan jaringan. Di dalam matematika, teori graf adalah cabang kajian yang mempelajari tentang sifat-sifat dari graf. Suatu graf G adalah pasangan himpunan (V, E) di mana V tidak kosong, dan E (mungkin kosong) adalah himpunan dari pasangan tak beraturan elemen V . Elemen V disebut simpul dan elemen dari E disebut sisi. Terkadang kita menulis $V(G)$ untuk simpul-simpul G dan $E(G)$ untuk sisi-sisi G . Simpul biasanya direpresentasikan sebagai titik dan sisi direpresentasikan sebagai garis yang menghubungkan simpul pada bidang. Sehingga, Graf G dapat ditulis sebagai $G = (V(G), E(G))$. Selain simpul dan sisi, berikut beberapa terminologi dasar lainnya dalam teori graf yang digunakan pada penelitian ini (Hartsfield & Ringel, 2013).

1. Jika a dan b merupakan simpul pada graf G , kita dapat mengatakan bahwa a bertetangga (*adjacent*) dengan b jika terdapat suatu sisi di antara a dan b .

2. Derajat (*degree*) pada suatu simpul adalah banyaknya sisi yang menempel dengan simpul tersebut. Suatu simpul yang memiliki derajat 0 disebut simpul yang terisolasi.
3. Suatu jalan (*walk*) pada graf G adalah suatu barisan berhingga ($W = v_0 e_1 v_1 e_2 v_2 \dots e_k v_k$) yang suku-sukunya bergantian antara simpul dan sisi. Simpul v_0 dan v_k berturut-turut disebut simpul awal dan simpul akhir dari W . Sedangkan simpul-simpul v_1, v_2, \dots, v_{k-1} disebut simpul-simpul internal dari W . Dan k disebut panjang (*length*) dari W .
4. Jika semua sisi $e_1, e_2, e_3, \dots, e_k$ dalam jalan W masing-masing berbeda, maka W disebut jejak (*trail*), dan jika semua simpul $v_0, v_1, v_2, \dots, v_k$ dalam jalan W juga masing-masing berbeda, maka W disebut suatu lintasan (*path*).
5. Berdasarkan orientasi arah pada sisi, maka graf dapat dibedakan menjadi dua jenis, yaitu graf berarah (*directed graph*) dan graf tak berarah (*undirected graph*). Graf berarah adalah graf yang setiap sisinya diberikan orientasi arah. Sebaliknya, graf tak berarah adalah graf yang setiap sisinya tidak diberikan orientasi arah. Pada graf tak berarah, tidak memperhatikan urutan pasangan simpul yang dihubungkan oleh sisi, dengan demikian pada himpunan E berlaku $(V_j, V_k) = (V_k, V_j)$. Sedangkan, pada graf berarah, himpunan E berlaku $(V_j, V_k) \neq (V_k, V_j)$.

2.2.7 Social Network Analysis

Istilah jaringan sosial telah menjadi hal yang cukup umum bagi publik karena perkembangan media sosial yang pesat, sehingga mendorong publik untuk menggunakan media sosial dan melakukan relasi secara tidak langsung ke orang lain. Selama bertahun-tahun, sosilog mencoba menemukan berbagai cara untuk mendefinisikan berbagai jenis relasi pada jaringan sosial ke bentuk analisis yang sistematis yang kemudian disebut analisis jaringan sosial atau *Social Network Analysis* (SNA). SNA terdiri dari seperangkat teknik metodologis yang bertujuan untuk menggambarkan dan mengeksplorasi pola-pola yang tampak dalam relasi sosial yang dibentuk oleh individu atau kelompok antara satu sama lain. Dalam arti yang lebih luas, jaringan sosial dibangun dari data relasional dan dapat didefinisikan sebagai perangkat entitas sosial, seperti individu, kelompok, atau organisasi, dengan beberapa pola hubungan atau interaksi di antara mereka. Struktur yang mendasari jaringan tersebut adalah objek studi pada SNA. Metode dan teknik SNA dirancang untuk menemukan pola interaksi antara aktor sosial di dalam jaringan. SNA telah menjadi pendekatan yang populer di berbagai bidang, salah satunya pada bidang bisnis. Perusahaan dapat menggunakan SNA untuk memaksimalkan informasi produk mereka menyebar dari percakapan antara pengguna (*word-of-mouth*) dengan menargetkan pengguna yang memiliki pengaruh yang lebih tinggi di dalam jaringan (Scott & Stokman, 2015).

Struktur dari jaringan sosial dapat direpresentasikan menggunakan pemanfaatan dari teori graf, di mana suatu jaringan sosial dapat dimodelkan menjadi suatu graf. Suatu graf terdiri dari dua komponen utama yaitu simpul dan sisi. Simpul dapat merepresentasikan entitas sosial, berupa individu atau kelompok, dan sisi merepresentasikan hubungan atau interaksi antara entitas sosial tersebut. Sisi dapat berarah atau tak berarah, tergantung apakah sifat relasinya asimetris atau simetris. Terdapat beberapa situasi di mana relasi yang dimiliki suatu jaringan sosial mempunyai orientasi arah, seperti relasi saling mengikuti atau *following* di antara pengguna media sosial. Terdapat juga beberapa situasi di mana relasi yang dimiliki suatu jaringan sosial tidak mempunyai orientasi arah, seperti relasi keberadaan atau tidak adanya interaksi percakapan atau pertemanan di antara beberapa aktor di jaringan. Sisi pada graf dapat mempunyai bobot atau tanpa bobot, kecuali dinyatakan secara eksplisit, selalu diasumsikan bahwa graf tidak mempunyai bobot. Graf tak berbobot adalah graf biner karena menilai

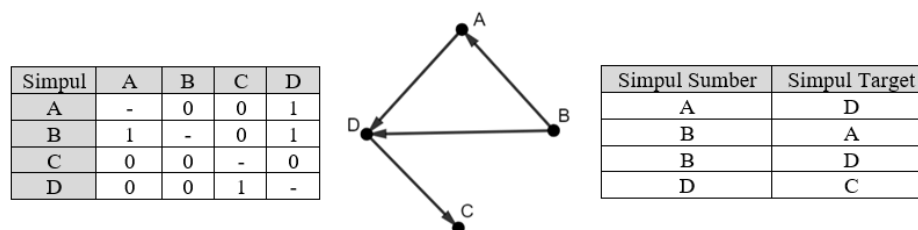
keberadaan sisi atau tidak adanya sisi di antara pasangan simpul di graf. Nilai bobot pada jaringan direpresentasikan dengan tebal atau tipis sisi yang menghubungkan pasangan simpul pada visualisasi jaringan, bobot tersebut merepresentasikan intensitas relasi pada sepasang aktor di jaringan (Oliveira & Gama, 2012).

Karena hubungan yang cukup erat antara SNA dan teori graf, untuk menghindari ambiguitas dan memperjelas bahasa yang diadopsi, pada Tabel 2.4 ditunjukkan terminologi yang digunakan pada teori graf dan SNA. Pada penelitian ini, penulis sering menggunakan terminologi tersebut secara bergantian.

Tabel 2.4 Terminologi pada Teori Graf dan SNA

No.	Teori Graf	SNA
1.	Graf	Jaringan
2.	Simpul	Aktor
3.	Sisi	Interaksi

Terdapat dua jenis struktur data yang umum digunakan untuk merepresentasikan suatu graf, yang pertama adalah struktur daftar yang biasa disebut *adjacency list* atau *edge list*, dan yang kedua adalah struktur matriks yang biasa disebut *adjacency matrix* atau matriks ketetanggaan. Kedua struktur data tersebut sesuai untuk menyimpan data pada graf dan selanjutnya menganalisis lebih lanjut. Struktur data *edge list* umum digunakan pada SNA karena cocok untuk menyimpan data jaringan yang besar dan jarang, sehingga mengurangi ruang penyimpanan yang diperlukan (Oliveira & Gama, 2012). Pada Gambar 2.1 menunjukkan contoh ilustrasi bagaimana graf dapat diwakili oleh *edge list* dan *adjacency matrix*.



Gambar 2.1 Representasi Graf dengan *Adjacency Matrix* dan *Edge List*

Terdapat beberapa pengukuran atau metrik yang populer digunakan pada SNA, metrik-metrik ini berguna dalam memberikan atau menambah wawasan mengenai struktur jaringan sosial, sebagai pelengkap ketika melihat jaringan hanya dengan berdasarkan representasi atau visualisasi secara grafis. Wawasan mengenai struktur jaringan sosial ini bertujuan untuk memahami pola interaksi sosial yang mendasari dari jaringan sosial tersebut. Pengukuran pada SNA dapat dibagi menurut tingkat analisis yang ingin dilakukan, pada tingkat unit aktor (lokal) atau pada tingkat keseluruhan jaringan (global). Pada tingkat unit aktor dilakukan pengukuran *centrality* sebagai cara untuk memahami bagaimana posisi simpul di letakkan pada keseluruhan struktur graf, sehingga, dapat membantu untuk mengidentifikasi aktor yang paling berpengaruh atau *key actor* di dalam jaringan sosial. Lalu, analisis *network properties* pada tingkat keseluruhan jaringan yang memberikan informasi atau penilaian tentang struktur dan sifat-sifat yang mendasari pada keseluruhan struktur jaringan sosial (Oliveira & Gama, 2012).

Centrality adalah ukuran tentang bagaimana posisi aktor diletakkan pada keseluruhan struktur jaringan sosial dan dapat dihitung berdasarkan beberapa metrik. Metrik yang paling umum digunakan adalah metrik *degree*, *betweenness*, *closeness*, dan *eigenvector centrality*. Metrik-metrik ini menentukan kepentingan relatif dari seorang aktor di dalam jaringan dan menunjukkan bagaimana interaksi terkonsentrasi pada beberapa aktor, sehingga dapat memberikan gambaran tentang kekuatan sosial mereka. Nilai *centrality* yang lebih tinggi

dikaitkan dengan aktor yang mempunyai kekuatan sosial di dalam jaringan, karena posisi sentral yang mereka miliki menawarkan beberapa keuntungan, seperti akses yang lebih mudah dan lebih cepat ke aktor lainnya dan kemampuan untuk melakukan kontrol atas aliran informasi antara aktor lainnya di dalam jaringan. Hal yang harus diperhitungkan adalah nilai dari hasil perhitungan beberapa metrik tingkat unit aktor ini mungkin perlu dinormalisasi untuk melakukan perbandingan pada setiap aktor di dalam jaringan (Oliveira & Gama, 2012).

Sebelum menjelaskan masing-masing metrik pada tingkat analisis keseluruhan jaringan, terdapat tiga konsep dasar yang harus diperkenalkan terlebih dahulu, yaitu lintasan (*path*), jarak geodesik (*geodesic distance*), dan eksentrisitas (*eccentricity*). Lintasan adalah urutan simpul di mana pasangan berurutan dari simpul yang tidak berulang dihubungkan oleh sisi. Selanjutnya, jarak geodesik dapat didefinisikan sebagai jarak lintasan terpendek (*shortest-path*) antara sepasang simpul. Lalu, eksentrisitas adalah jarak geodesik terbesar antara suatu simpul tertentu dengan simpul lainnya di dalam jaringan. Ketiga konsep ini merupakan dasar bagi sebagian besar metrik untuk analisis tingkat keseluruhan jaringan, yaitu *density*, *modularity*, *diameter*, *average path length*, *average degree*, dan *connected components* (Oliveira & Gama, 2012).

Pada Tabel 2.3 menunjukkan dua bagian pengukuran berdasarkan analisis tingkat keseluruhan jaringan, yaitu *network properties*, dan analisis tingkat aktor, yaitu *centrality*, beserta metrik yang digunakan pada setiap pengukurannya.

Tabel 2.5 Pengukuran pada SNA Berdasarkan Tingkat Analisis

No.	Tingkat Analisis	Pengukuran	Metrik
1.	Keseluruhan jaringan	<i>Network Properties</i>	<i>Size & Order</i>
			<i>Density</i>
			<i>Diameter</i>
			<i>Modularity</i>
			<i>Average Path Length</i>
			<i>Average Degree</i>
			<i>Connected Components</i>
2.	Unit Aktor	<i>Centrality</i>	<i>Degree</i>
			<i>Betweenness</i>
			<i>Closeness</i>
			<i>Eigenvector</i>

Menganalisis suatu struktur jaringan berbeda dengan analisis statistika sederhana lainnya, karena dibutuhkan konteks mengenai model dan elemen dari struktur jaringan tersebut. Sehingga, pada SNA dibutuhkan juga visualisasi jaringan sebagai pelengkap informasi, baik visualisasi jaringan secara *diagrammatic* (untuk jaringan yang kecil dan tingkat analisis lokal) atau *topological* (untuk mendeteksi pola struktur pada keseluruhan jaringan). Namun, visualisasi terkadang sulit dibaca atau dianalisis ketika model jaringan berukuran besar atau kompleks, sehingga dibutuhkan pengukuran yang dapat menggambarkan model dan elemen dari suatu jaringan secara kontekstual dan holistik. Metrik-metrik pada setiap pengukuran yang digunakan di SNA saling melengkapi satu sama lain, sehingga wawasan mengenai model jaringan dapat direpresentasikan secara lebih kontekstual dan holistik (Grandjean, 2021).

Misalnya, pada pengukuran *network properties*, perhitungan metrik paling sederhana yang dapat dilakukan pertama kali adalah membandingkan jumlah simpul (*order*) dan sisi (*size*) dari dua jaringan yang berbeda. Di antara kedua metrik yang sederhana tersebut, kerapatan jaringan (*density*) dapat menjadi suatu indikator yang baik ketika membandingkan jaringan yang berukuran besar, di mana tidak memungkinkan untuk menghitung semua elemen secara visual. *Diameter* dapat digunakan untuk menggambarkan bagaimana *density* didistribusikan, metrik ini juga dapat diterjemahkan sebagai lebar dari jaringan. Metrik *average path length* dapat

berfungsi sebagai pelengkap metrik *diameter*, karena nilai dari *diameter* dapat dipengaruhi oleh beberapa simpul yang jauh dari komponen utama graf. Dengan alasan yang hampir sama, metrik *average degree* juga dapat berfungsi sebagai pelengkap metrik *density*, karena nilai dari *density* dapat dipengaruhi oleh beberapa sisi yang jauh dari komponen utama graf. Metrik *connected components* adalah perhitungan tingkat global sederhana yang berfungsi untuk membedakan jaringan di mana semua komponen terhubung atau terputus, metrik ini dapat memberikan informasi mengenai komponen utama di dalam jaringan. Sedangkan, metrik *modularity* berfungsi untuk menemukan struktur komunitas atau kelompok (*community detection*) yang terbentuk di dalam jaringan. Dalam beberapa kasus, menarik untuk membandingkan komponen dan kelompok yang terbentuk pada suatu jaringan, sehingga metrik *modularity* dapat berfungsi sebagai pelengkap metrik *connected components* (Grandjean, 2021).

Sedangkan, pada pengukuran *centrality*, pengukuran paling sederhana yang dapat dilakukan pertama kali adalah metrik *degree centrality*, karena hanya menghitung jumlah interaksi yang dilakukan antara suatu aktor ke aktor lainnya di dalam jaringan, metrik ini mengidentifikasi aktor yang memiliki *social connections* yang paling banyak di dalam jaringan. Metrik *betweenness centrality* dapat menjadi pelengkap metrik *degree centrality*, karena metrik ini dapat mengungkapkan elemen struktural yang menjadi jembatan di antara kelompok-kelompok yang besar di dalam jaringan. Metrik *closeness centrality* dapat menjadi pelengkap metrik *betweenness centrality*, karena metrik ini dapat mengungkapkan elemen yang mempunyai posisi sentral secara geografis di dalam jaringan. Lalu, metrik *eigenvector centrality* dapat menjadi pelengkap metrik-metrik *centrality* lainnya, karena pada metrik *eigenvector centrality*, selain menghitung nilai sentralitas pada suatu aktor, metrik ini juga menghitung nilai sentralitas aktor-aktor lainnya yang terhubung dengan aktor tersebut (Grandjean, 2021).

Dua jaringan sosial yang mempunyai ukuran yang identik, mungkin saja tidak memiliki hasil pengukuran yang sama, karena tergantung pada struktur internal atau topologi jaringannya. Hal ini mengarah pada kombinasi pengukuran struktur model jaringan dan metrik yang digunakan. Berdasarkan jenis pengukurannya, metrik-metrik pada SNA dapat dibagi menjadi tiga bagian besar, yaitu *connections*, *distributions*, dan *segmentation*. *Connections* mengukur bagaimana aktor saling terhubung satu sama lain dan struktur jaringan sosial yang terbentuk berdasarkan koneksi antara aktor tersebut. *Distributions* mengukur bagaimana alur informasi menyebar dan bagaimana aktor diletakkan atau didistribusikan di dalam jaringan sosial. *Segmentation* mengukur bagaimana segmentasi jaringan sosial ke dalam kelompok-kelompok atau komponen-komponen dan ikatan kohesif antara aktor di dalam kelompok atau komponen tersebut. Tiga jenis pengukuran beserta metrik-metrik sebagai representasi pada setiap pengukurannya ditunjukkan pada Tabel 2.6 (Alamsyah dkk., 2014).

Tabel 2.6 Pengukuran SNA Berdasarkan Jenis Analisis

No.	Pengukuran	Deskripsi	Metrik
1.	<i>Connections</i>	Mengukur bagaimana aktor terhubung satu sama lain dan struktur jaringan sosial yang terbentuk.	<i>Size</i>
			<i>Order</i>
			<i>Density</i>
			<i>Average Degree</i>
2.	<i>Distributions</i>	Mengukur bagaimana alur informasi menyebar dan bagaimana aktor didistribusikan di dalam jaringan.	<i>Diameter</i>
			<i>Average Path Length</i>
			<i>Centrality</i>
3.	<i>Segmentation</i>	Mengukur segmentasi jaringan ke dalam kelompok-kelompok dan ikatan kohesif antara aktor di dalam kelompok tersebut.	<i>Modularity</i>
			<i>Connected Components</i>

Setelah melakukan perhitungan pada seluruh metrik-metrik SNA, seluruh nilai hasil perhitungan tersebut akan dilakukan perbandingan. Pada analisis tingkat keseluruhan jaringan, perbandingan akan dilakukan di antara nilai-nilai metrik *network properties* pada kedua jaringan sosial. Jaringan sosial yang mempunyai nilai metrik *network properties* yang lebih unggul, mempunyai makna bahwa jaringan sosial tersebut lebih aktif dan mempunyai interaksi antara aktor yang lebih baik. Sedangkan, pada analisis tingkat unit aktor, perbandingan akan dilakukan di antara nilai-nilai metrik *centrality* pada setiap aktor di masing-masing jaringan sosial. Aktor yang mempunyai nilai yang tertinggi di setiap metrik *centrality* merupakan aktor yang berpengaruh atau *key actor* di dalam jaringan sosial tersebut (Alamsyah dkk., 2014).

2.2.7.1 Network Properties

SNA memiliki beberapa metrik pada pengukuran *network properties* yang berguna untuk menganalisis struktur jaringan sosial secara keseluruhan dan mendeskripsikan struktur jaringan sosial tersebut. Beberapa metrik pada pengukuran *network properties* adalah *order*, *size*, *density*, *modularity*, *diameter*, *average path length*, *average degree*, dan *connected components*.

2.2.7.1.1 Order dan Size

Pada suatu jaringan, *order* adalah banyaknya simpul, dan *size* adalah banyaknya sisi pada jaringan tersebut. Nilai *order* dan *size* pada suatu jaringan sosial menunjukkan banyaknya aktor yang berinteraksi pada jaringan sosial tersebut. Formula untuk menghitung nilai *size* dan *order* adalah sebagai berikut (Oliveira & Gama, 2012).

$$\begin{aligned} m &= |E(G)| \\ n &= |V(G)| \end{aligned} \quad (2.1)$$

keterangan:

- $E(G)$: himpunan sisi pada suatu graf G
- $V(G)$: himpunan simpul pada suatu graf G

Pada pembahasan berikutnya, banyaknya simpul akan direpresentasikan sebagai variabel n dan banyaknya sisi direpresentasikan sebagai variabel m (Oliveira & Gama, 2012).

2.2.7.1.1 Density

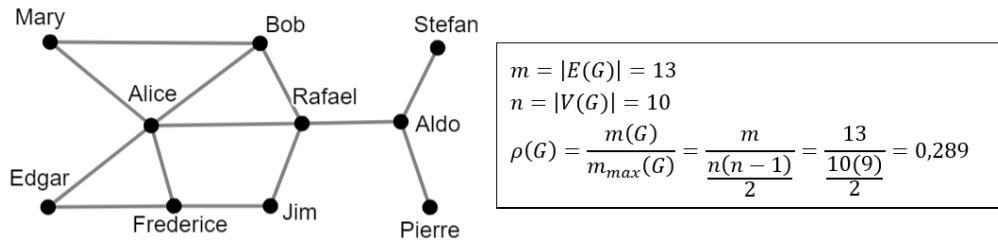
Density (ρ) atau kerapatan jaringan merupakan perhitungan banyaknya sisi yang ada dibandingkan dengan banyaknya sisi maksimum yang mungkin ada di dalam suatu jaringan. *Density* mampu menjelaskan tingkat keterhubungan secara umum di dalam suatu jaringan. Semakin tinggi nilai *density*, maka semakin baik karena menggambarkan bahwa aktor-aktor yang ada di dalam jaringan sosial lebih saling terhubung antara satu sama lain. *Density* adalah besaran yang berubah dari minimum 0, ketika jaringan sosial tidak mempunyai interaksi sama sekali, dan maksimum 1, ketika jaringan sosial terhubung secara sempurna. Oleh karena itu, nilai *density* yang tinggi dikaitkan dengan jaringan yang padat, dan nilai *density* yang rendah dikaitkan dengan jaringan yang jarang. Formula untuk menghitung *density* adalah sebagai berikut (Oliveira & Gama, 2012).

$$\rho(G) = \frac{m(G)}{m_{\max}(G)} = \frac{m}{\frac{n(n-1)}{2}}, \quad 0 < \rho < 1 \quad (2.2)$$

keterangan:

- m : jumlah sisi pada jaringan
- n : jumlah simpul pada jaringan

Sebagai contoh ilustrasi untuk cara menghitung nilai *density*, pada Gambar 2.3 ditunjukkan suatu jaringan sosial, beserta proses menghitung metrik *density* dari jaringan sosial tersebut.



Gambar 2.2 Contoh Ilustrasi Perhitungan Metrik *Density*

2.2.7.1.2 Modularity

Modularity merupakan metrik yang umum digunakan untuk mendeteksi kelompok atau mengetahui kualitas pembagian jaringan ke dalam kelompok. Ide dasar dari *modularity* adalah bahwa suatu jaringan menunjukkan struktur kelompok yang berarti jika jumlah sisi di antara kelompok lebih sedikit daripada yang diharapkan, berdasarkan pilihan acak. Dengan asumsi ini, semakin tinggi nilai *modularity*, maka semakin baik partisinya, yang berarti kelompok yang ditemukan terhubung erat secara internal dan jarang terhubung secara eksternal. Formula untuk mendapatkan nilai *modularity* adalah sebagai berikut (Oliveira & Gama, 2012).

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{i,j} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j) \quad (2.3)$$

keterangan:

- $A_{i,j}$: matriks ketetanggaan antara simpul i dan j
- k_i : derajat pada simpul i
- m : jumlah sisi pada jaringan
- c_i : kelompok i dimana simpul i dan j berada
- $\delta(c_i, c_j)$: *kronecker delta*

Berdasarkan formula di atas, $\frac{k_i k_j}{2m}$ merupakan jumlah ekspektasi sisi-sisi yang ada di antara kedua simpul tersebut. Maka dapat terlihat bahwa $Q \in [-1,1]$, jika Q positif, maka ada kemungkinan menemukan struktur kelompok di dalam jaringan, jika Q tidak hanya positif, tetapi juga besar, maka partisi tersebut dapat mencerminkan struktur kelompok yang sebenarnya. Pada praktiknya, nilai *modularity* sekitar 0,3 dapat menunjukkan indikator yang baik dari keberadaan struktur kelompok yang berarti (Oliveira & Gama, 2012).

Mengikuti alasan ini, diketahui bahwa semakin tinggi nilai *modularity*, maka semakin baik pembagian jaringan yang diperoleh. Sehingga, pendekatan yang ingin dicapai adalah memaksimalkan nilai pada metrik ini, dengan menghitung setiap kemungkinan partisi yang ada pada jaringan dan memilih partisi yang memberikan nilai *modularity* yang tertinggi. Untuk memaksimalkan nilai *modularity* secara efisien, maka dapat dilakukan dengan algoritma Louvain. Algoritma Louvain adalah suatu algoritma *community detection* yang secara rekursif menggabungkan kelompok menjadi satu simpul dan mengeksekusi *modularity clustering* pada jaringan tersebut yang telah diringkas. Algoritma Louvain terdiri dari dua tahapan yaitu *modularity optimization* dan *community aggregation* (Barabási, 2016).

1. Modularity Optimization

Pertama, pada tahapan ini secara acak mengurutkan seluruh simpul pada jaringan. Kemudian, satu per-satu simpul dipindahkan ke kelompok yang berbeda, dimisalkan kelompok C . Simpul akan terus dipindahkan ke kelompok lain jika terdapat peningkatan nilai *modularity*. Jika tidak terdapat peningkatan nilai *modularity*, maka simpul tersebut tetap

berada di kelompok aslinya. Proses ini terus-menerus diterapkan ke seluruh simpul sampai tidak ada lagi peningkatan yang signifikan pada nilai *modularity*. Perubahan nilai *modularity* (ΔQ) dapat dihitung dengan formula sebagai berikut (Barabási, 2016).

$$\Delta Q = \left[\frac{\sum_{in} + 2k_{i,in}}{2m} - \left(\frac{\sum_{tot} + k_i}{2m} \right)^2 \right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m} \right)^2 - \left(\frac{k_i}{2m} \right)^2 \right] \quad (2.4)$$

keterangan:

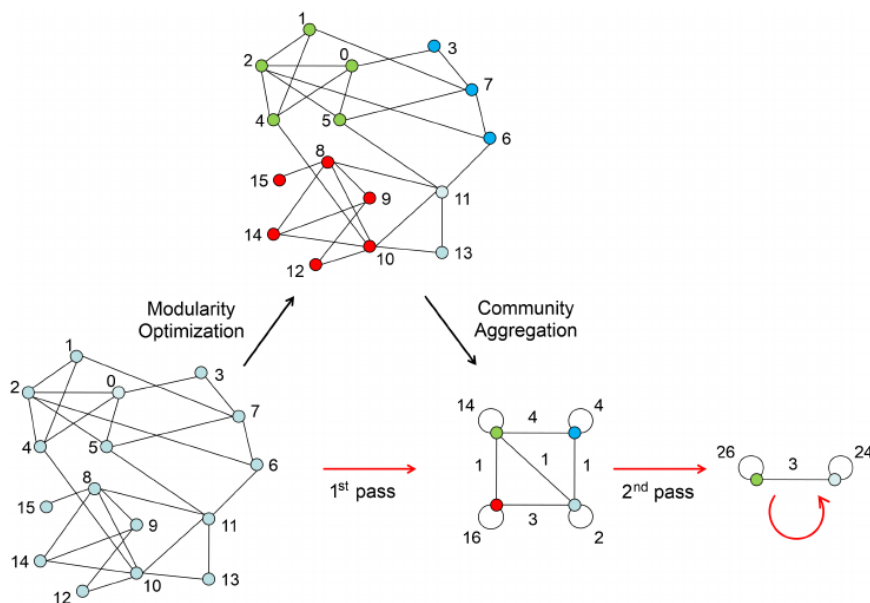
- \sum_{in} : jumlah sisi di dalam kelompok C
- \sum_{tot} : jumlah sisi yang menempel dengan simpul yang ada di kelompok C
- k_i : jumlah sisi yang menempel pada simpul i
- $k_{i,in}$: jumlah sisi pada simpul i yang menempel dengan simpul yang ada di kelompok C
- m : jumlah sisi pada jaringan

2. Community Aggregation

Setelah menyelesaikan tahapan pertama, semua simpul pada kelompok yang sama akan digabung menjadi satu simpul raksasa. Tahapan ini menghasilkan *output* berupa simpul yang *self-loop* yang di mana bobotnya merupakan jumlah dari semua bobot simpul yang ada pada kelompok, dengan kata lain, simpul-simpul yang ada telah diringkas menjadi hanya satu simpul raksasa.

Jumlah kelompok akan terus berkurang pada setiap iterasi atau bisa disebut juga sebagai *pass*. *Pass* akan terus diulang pada masing-masing tahapan algoritma Louvain sampai tidak ada lagi perubahan secara signifikan dan nilai *modularity* maksimum telah tercapai (Barabási, 2016).

Kelompok-kelompok yang terbentuk pada proses perhitungan metrik *modularity* bisa disebut juga sebagai *modularity class*. Setiap aktor di dalam jaringan merupakan anggota dari salah satu kelompok tersebut. Setiap kelompok, setidaknya memiliki dua atau lebih aktor sebagai anggota kelompok. Kelompok yang memiliki nilai presentase, jumlah aktor sebagai anggota kelompok dibandingkan dengan jumlah aktor di seluruh jaringan, yang lebih besar atau sama dengan 1 %, maka kelompok tersebut dinamakan kelompok yang besar (Muff dkk., 2005).



Gambar 2.3 Ilustrasi Algoritma Louvain

<pre> 1: function LouvainAlgorithm(Graph G) 2: G' = G 3: C = the index of community of each nodes of G' 4: Initialize each nodes with its own community 5: q = -∞ 6: while q < Q(G', C) do 7: q = Q(G', C) 8: C' = MoveNodes(G') //Phase 1 9: G' = Aggregate(G', C) //Phase 2 10: C = put each node of G' in its own community 11: end while 12: return G' 13: end function </pre>	<pre> 14: function MoveNodes(Graph G) 15: C = the index of communities for each nodes of G 16: while one or more nodes are moved do 17: for random v ∈ V(G) do 18: best_q = -∞ 19: best_c = community of v 20: for all neighboring nodes n of v do 21: gain_q = ΔQ between v and n 22: if best_q < gain_q < then 23: best_q = gain_q 24: best_c = community of n 25: end if 26: end for 27: C = Place v in the best_q 28: end for 29: end while 30: return C 31: end function 32: function Aggregate(Graph G, Partition C) 33: G' = aggregate nodes which are in same community based on C 34: return G' </pre>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Gambar 2.4 Pseudocode Algoritma Louvain

2.2.7.1.3 Diameter

Diameter adalah jarak lintasan terpendek (*shortest-path*) terpanjang atau terjauh antara sepasang aktor di dalam jaringan. Semakin rendah nilai *diameter* maka semakin baik, karena artinya proses penyebaran informasi antara suatu aktor dengan aktor lainnya, dengan jarak yang terjauh, hanya perlu melewati sedikit aktor. Nilai *diameter* yang lebih rendah mengindikasikan proses penyebaran informasi akan memakan waktu yang lebih sedikit karena melibatkan lebih sedikit aktor. Pada jaringan yang memiliki kerapatan jaringan yang lebih rendah, umumnya memiliki nilai *diameter* yang lebih besar daripada jaringan dengan *adjacency matrix* yang lebih penuh, karena memiliki lintasan yang lebih sedikit pada setiap pasang simpul. Formula untuk mendapatkan *diameter* pada suatu jaringan adalah sebagai berikut (Oliveira & Gama, 2012).

$$D = \max \{d_{v_i, v_j} : v \in V\} \quad (2.5)$$

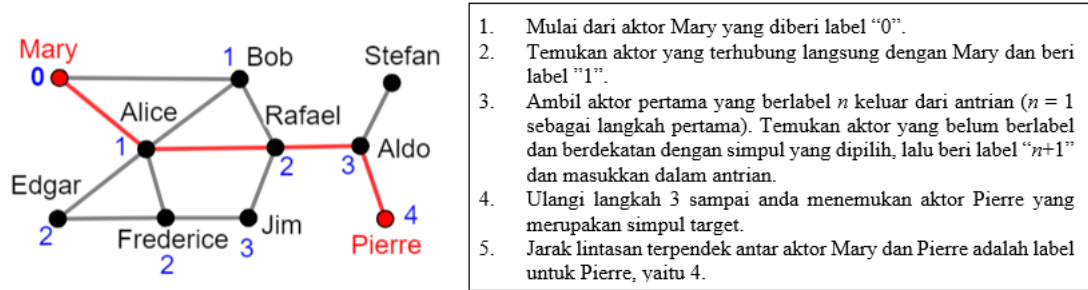
keterangan:

- V : himpunan simpul di dalam jaringan
- d_{v_i, v_j} : jarak lintasan terpendek antara simpul i dan j

Pada jaringan yang berukuran besar, lintasan terpendek dapat ditentukan menggunakan algoritma *Breadth-First Search* (BFS). BFS adalah algoritma penjelajahan graf (*graph traversal*) yang menelusuri graf dari simpul akar (*root node*) dan menjelajahi semua simpul tetangga. Kemudian, algoritma ini memilih simpul terdekat dan menjelajahi semua simpul yang belum dijelajahi. Saat menggunakan algoritma BFS untuk penjelajahan, setiap simpul dalam graf dapat dianggap sebagai simpul akar. BFS dapat dimanfaatkan untuk menentukan jarak lintasan terpendek pada graf. Identifikasi jarak lintasan terpendek antara simpul i dan simpul j dengan algoritma BFS mengikuti langkah-langkah berikut (Barabási, 2016):

1. Mulai dari simpul i yang kita beri label "0".
2. Temukan simpul yang terhubung langsung ke i . Beri label "1" dan tempatkan di dalam antrian.
3. Ambil simpul pertama yang berlabel n keluar dari antrian ($n = 1$ sebagai langkah pertama). Temukan simpul yang belum berlabel dan berdekatan dengan simpul yang dipilih, lalu beri label " $n+1$ " dan masukkan dalam antrian.
4. Ulangi langkah 3 sampai anda menemukan simpul j yang merupakan simpul target atau tidak ada lagi simpul di dalam antrian.
5. Jarak antar i dan j adalah label untuk j . Jika j tidak mempunyai label, maka $d_{v_i, v_j} = \infty$.

Sebagai contoh ilustrasi algoritma BFS, berikut adalah contoh suatu jaringan sosial, penulis akan mencari jarak lintasan terpendek dari aktor Mary dengan aktor Pierre menggunakan algoritma BFS. Gambar 2.5 menunjukkan ilustrasi implementasi algoritma BFS pada suatu jaringan.



Gambar 2.5 Contoh Implementasi Algoritma BFS pada Jaringan

2.2.7.1.4 Average Path Length

Average path length merupakan perhitungan rata-rata jarak lintasan terpendek di antara setiap pasang simpul yang ada di dalam suatu jaringan. Semakin kecil nilai *average path length*, maka semakin baik, karena rata-rata jarak yang harus ditempuh untuk melakukan penyebaran informasi lebih pendek sehingga arus persebaran informasi di jaringan lebih efisien. Kita dapat menggunakan algoritma BFS, seperti yang sudah dibahas sebelumnya, untuk menentukan jarak lintasan terpendek di dalam jaringan. Formula untuk menghitung *average path length* adalah sebagai berikut (Oliveira & Gama, 2012).

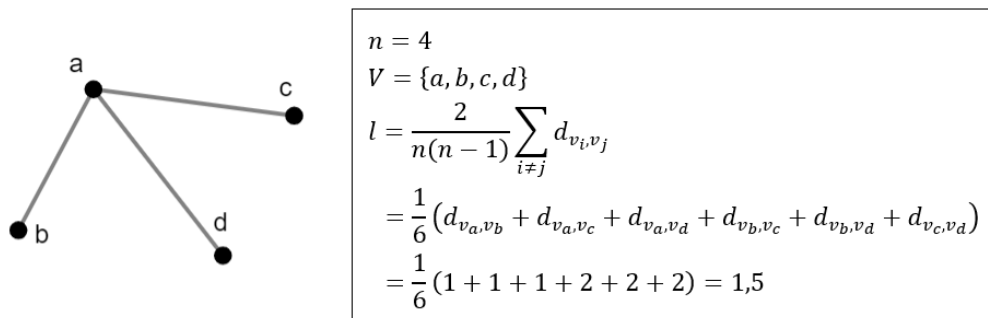
$$l = \frac{2}{n(n-1)} \sum_{i \neq j} d_{v_i, v_j} \quad (2.6)$$

keterangan:

n : jumlah simpul pada jaringan

d_{v_i, v_j} : jarak lintasan terpendek antara simpul i dan j

Ketika terdapat kasus di mana jaringan memiliki lebih dari satu *connected components*, maka formula di atas tidak berlaku, karena jarak lintasan terpendek secara konvensional didefinisikan menjadi tak hingga ketika tidak ada lintasan yang menghubungkan dua simpul. Dalam situasi seperti ini, lebih tepat untuk menghitung rata-rata berdasarkan jumlah jarak lintasan terpendek pada setiap simpul di jaringan dibagi dengan banyaknya lintasan terpendek pada setiap simpul di jaringan. Sebagai contoh ilustrasi untuk cara menghitung nilai *average path length*, pada Gambar 2.6 ditunjukkan suatu graf dan proses menghitung dan nilai *average path length* dari graf tersebut.



Gambar 2.6 Contoh Ilustrasi Perhitungan Metrik *Average Path Length*

2.2.7.1.5 Average Degree

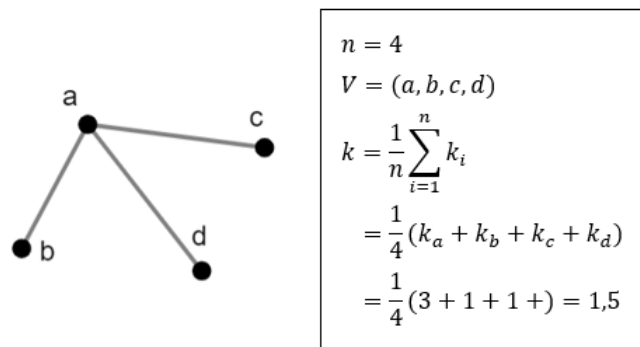
Average degree merupakan perhitungan rata-rata derajat pada setiap simpul atau rata-rata banyaknya sisi yang menghubungkan suatu simpul ke simpul lain di dalam jaringan. Semakin besar nilai *average degree* yang dimiliki oleh jaringan maka semakin baik, karena apabila suatu aktor dapat menyebarkan informasi ke lebih banyak aktor lainnya, maka akan mempercepat penyebaran informasi di dalam jaringan sosial. Formula untuk menghitung *average degree* adalah sebagai berikut (Oliveira & Gama, 2012).

$$k = \frac{1}{n} \sum_{i=1}^n k_i \quad (2.7)$$

keterangan:

- n : jumlah simpul di jaringan
 k_i : derajat pada simpul ke- i

Sebagai contoh ilustrasi cara untuk menghitung nilai *average degree*, pada Gambar 2.7 ditunjukkan suatu graf dan proses menghitung metrik *average degree* dari graf tersebut.



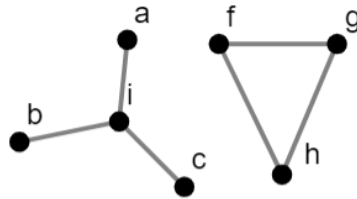
Gambar 2.7 Contoh Ilustrasi Perhitungan Metrik *Average Degree*

2.2.7.1.6 Connected Components

Metrik *connected components* merupakan perhitungan banyaknya bagian-bagian yang terpisah atau tidak terkoneksi dengan keseluruhan jaringan. Semakin kecil nilai *connected components* maka semakin baik, karena artinya keseluruhan aktor tidak banyak terpisah ke dalam kelompok-kelompok kecil yang tidak saling terkoneksi. *Connected components* dapat dicari dengan menggunakan algoritma BFS yang telah dibahas sebelumnya. Identifikasi jumlah *connected components* dengan algoritma BFS dapat mengikuti langkah-langkah berikut (Barabási, 2016).

1. Mulai dari simpul i yang dipilih secara acak dan lakukan algoritma BFS, beri label semua simpul yang dicapai dengan $n = 1$.
2. Jika jumlah total simpul yang berlabel sama dengan total seluruh simpul di jaringan (N), maka jaringan tersebut terhubung, jika jumlah simpul yang berlabel lebih kecil daripada N , jaringan terdiri dari beberapa komponen. Untuk mengidentifikasi komponen-komponen tersebut, lanjutkan ke langkah tiga.
3. Tingkatkan label menjadi $n \rightarrow n + 1$. Pilih simpul j yang belum ditandai, beri label n . Gunakan BFS untuk menemukan semua simpul yang dapat dijangkau dari j , beri label semuanya dengan n . Kembali ke langkah 2.

Pada Gambar 2.8 ditunjukkan contoh ilustrasi dua *connected components* pada suatu graf dan *pseudocode* algoritma untuk mencari banyaknya *connected components* di dalam jaringan.



Algorithm *Number of Connected Components*

```

function Count_Connected_Components(graf G)
  count = 0
  for each vertex v in N do
    flag[v] = -1
  end for
  for (int v = 0, v < N, v++) do
    if flag[v] == -1 do
      BFS(v, flag)
      count++
    end if
  end for
  return count
end function

function BFS(v, flag)
  flag[v] = 1
  for each adjacent node u of v do
    if flag[u] == -1 do
      BFS(u, flag)
    end if
  end for
end function

```

Gambar 2.8 Contoh Ilustrasi dan *Pseudocode* Algoritma *Connected Components*

Rangkuman keseluruhan metrik *network properties* yang telah dijelaskan pada pembahasan sebelumnya, ditunjukkan pada Tabel 2.4.

Tabel 2.7 Rangkuman Metrik *Network Properties*

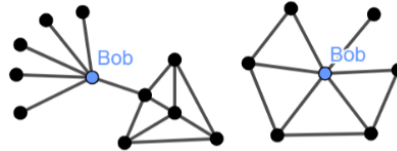
No.	<i>Network Properties</i>	Deskripsi	Perhitungan
1.	<i>Order</i>	Jumlah simpul di jaringan.	$m = E(G) $
2.	<i>Size</i>	Jumlah sisi di jaringan.	$n = V(G) $
3.	<i>Density</i>	Kerapatan pada jaringan.	$\rho(G) = \frac{m(G)}{m_{max}(G)} = \frac{2m}{n(n-1)}$
4.	<i>Modularity</i>	Kualitas pembagian jaringan ke dalam kelompok.	$Q = \frac{1}{2m} \sum_{i,j} \left[A_{i,j} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j)$
5.	<i>Diameter</i>	Jarak lintasan terpendek terpanjang di antara sepasang simpul di jaringan.	$D = \max \{d_{v_i, v_j} : v \in V\}$
6.	<i>Average Path Length</i>	Rata-rata jarak lintasan terpendek di setiap pasang simpul di jaringan.	$l = \frac{2}{n(n-1)} \sum_{i \neq j} d_{v_i, v_j}$
7.	<i>Average Degree</i>	Rata-rata derajat pada setiap simpul di jaringan.	$k = \frac{1}{n} \sum_{i=1}^n k_i$
8.	<i>Connected Components</i>	Jumlah kelompok simpul yang tidak terkoneksi dengan keseluruhan jaringan.	Algoritma BFS

2.2.7.2 Centrality

Setelah metrik *network properties*, metrik selanjutnya adalah *centrality*. Pengukuran *centrality* berguna untuk menganalisis pada tingkatan unit aktor yang bertujuan untuk mengidentifikasi aktor yang paling berpengaruh (*key actor*) di antara semua aktor lain yang ada di dalam suatu jaringan sosial. Terdapat empat metrik *centrality*, yaitu: *degree centrality*, *betweenness centrality*, *closeness centrality*, dan *eigenvector centrality*.

2.2.7.2.1 Degree Centrality

Metrik *degree centrality* menggambarkan ukuran *social connections* yang dipunyai aktor di dalam jaringan. Suatu aktor dengan nilai *degree centrality* yang tinggi bisa jadi mempunyai posisi yang sentral di dalam jaringan, namun juga, bisa saja berada jauh di tepi jaringan. Sebagai contoh, pada Gambar 2.9 menunjukkan kedua aktor bernama Bob yang memiliki nilai *degree centrality* yang tertinggi di masing-masing jaringan, akan tetapi, peran yang mereka mainkan sangat berbeda. Bob yang berada di jaringan sebelah kanan sangat sentral, sedangkan Bob yang berada di jaringan sebelah kiri sedikit berada di tepi (Fornito dkk., 2016).



Gambar 2.9 Dua Aktor dengan Nilai *Degree Centrality* yang Sama

Berikut adalah formula *degree centrality* untuk simpul i (Fornito dkk., 2016):

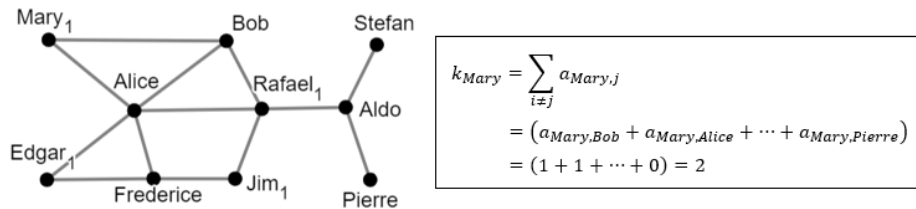
$$C_D(i) = k_i = \sum_{i \neq j} a_{v_i v_j}, \quad 0 < k_i < n \quad (2.8)$$

$$a_{v_i v_j} = \begin{cases} 1, & \text{apabila terdapat sisi di antara simpul } i \text{ dan } j \\ 0, & \text{yang lainnya} \end{cases}$$

keterangan:

n : jumlah simpul di jaringan.

Sebagai contoh ilustrasi untuk cara menghitung nilai *degree centrality* pada suatu aktor di dalam jaringan, pada Gambar 2.10 ditunjukkan suatu jaringan dan proses menghitung nilai *degree centrality* dari suatu aktor pada jaringan tersebut.



Gambar 2.10 Contoh Ilustrasi Perhitungan Metrik *Degree Centrality*

2.2.7.2.2 Betweenness Centrality

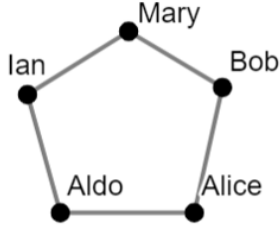
Gagasan pada metrik *betweenness centrality* adalah suatu metrik *centrality* yang tidak mementingkan seberapa banyak *social connections* yang dimiliki aktor, tetapi lebih di mana posisi aktor tersebut terletak di dalam jaringan. Aktor dengan nilai *betweenness centrality* tertinggi menempati peran penting dalam struktur jaringan, karena memiliki posisi di dalam jaringan yang memungkinkan aktor tersebut mengendalikan arus informasi dengan aktor maupun kelompok lainnya. Sebagai contoh, pada Gambar 2.8, aktor bernama Bob yang berada di jaringan sebelah kiri merupakan aktor yang memungkinkan untuk informasi dapat diteruskan dari kelompok kanan ke kelompok kiri, dan juga sebaliknya. Formula untuk menghitung nilai metrik *betweenness centrality* pada simpul i adalah sebagai berikut (Fornito dkk., 2016).

$$C_B(i) = g(i) = \frac{2}{(n-1)(n-2)} \cdot \sum_{h \neq i, h \neq j, j \neq i} \frac{\sigma_{hj}(i)}{\sigma_{hj}} \quad (2.9)$$

keterangan:

- $\sigma_{hj}(i)$: jumlah lintasan terpendek antara simpul h dan simpul j yang melalui simpul i
 σ_{hj} : jumlah lintasan terpendek antara simpul h dan j
 n : jumlah simpul di jaringan

Sebagai contoh ilustrasi, pada Gambar 2.11 ditunjukkan suatu jaringan dan proses menghitung nilai *betweenness centrality* dari suatu aktor pada jaringan tersebut.



$$\begin{aligned}
 n &= 5 \\
 C_B(Mary) &= \frac{2}{(n-1)(n-2)} \cdot \sum \frac{\sigma_{hj}(Mary)}{\sigma_{hj}} \\
 &= \frac{2}{(5-1)(5-2)} \left(\frac{\sigma_{Ian,Bob}(Mary)}{\sigma_{Ian,Bob}} + \frac{\sigma_{Ian,Alice}(Mary)}{\sigma_{Ian,Alice}} + \dots + \frac{\sigma_{Alice,Aldo}(Mary)}{\sigma_{Alice,Aldo}} \right) \\
 &= \frac{1}{6} \cdot \left(\frac{1}{1} + \frac{0}{1} + \frac{0}{1} + \frac{0}{1} + \frac{0}{1} + \frac{0}{1} \right) = 0,17
 \end{aligned}$$

Gambar 2.11 Contoh Ilustrasi Perhitungan Metrik *Betweenness Centrality*

2.2.7.2.3 Closeness Centrality

Metrik *closeness centrality* adalah pengukuran posisi dari keseluruhan aktor di dalam jaringan, gagasan pada metrik *closeness centrality* adalah suatu perhitungan untuk mencari aktor yang paling dekat dengan semua aktor lainnya, sehingga dapat diidentifikasi aktor yang paling cepat menyebarkan informasi ke seluruh aktor di dalam jaringan. Perhitungan metrik *closeness centrality* untuk suatu simpul merupakan *inverse* dari rata-rata jarak lintasan terpendek dari simpul tersebut ke setiap simpul lainnya di dalam jaringan. Oleh karena itu, biasanya metrik ini menghitung pada simpul dalam komponen terbesar di jaringan. Formula untuk menghitung nilai *closeness centrality* pada simpul i adalah sebagai berikut (Fornito dkk., 2016).

$$C_c(i) = \frac{n-1}{\sum_{i \neq j} d_{v_i} d_{v_j}} \quad (2.10)$$

keterangan:

- $d_{v_i} d_{v_j}$: jarak lintasan terpendek antara simpul i dan j
 n : jumlah simpul di jaringan

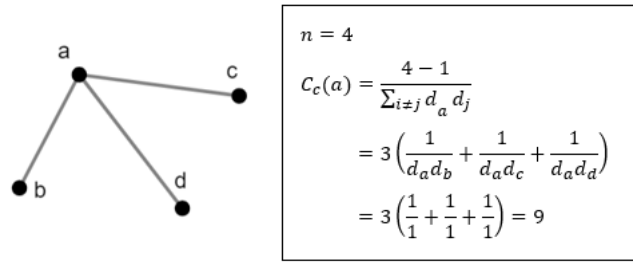
Pada jaringan dengan jumlah *connected components* lebih dari satu, maka dapat menggunakan formula yang disempurnakan oleh Wasserman dan Faust, dengan formulanya adalah sebagai berikut (NetworkX, 2022).

$$C_{c_{wf}}(i) = \frac{n-1}{N-1} \cdot \frac{n-1}{\sum_{i \neq j} d_{v_i} d_{v_j}} \quad (2.11)$$

keterangan:

- $d_{v_i} d_{v_j}$: jarak lintasan terpendek antara simpul i dan j
 n : jumlah simpul yang dapat dijangkau oleh simpul i
 N : jumlah simpul di jaringan

Sebagai contoh ilustrasi, pada Gambar 2.12 ditunjukkan suatu jaringan dan proses menghitung nilai *closeness centrality* dari suatu aktor pada jaringan tersebut.



Gambar 2.12 Contoh Ilustrasi Perhitungan Metrik *Closeness Centrality*

2.2.7.2.4 *Eigenvector Centrality*

Gagasan dari metrik *eigenvector centrality* adalah suatu perhitungan yang memperhatikan kualitas koneksi yang dimiliki suatu aktor di dalam jaringan, dengan kata lain, metrik ini mempertimbangkan derajat dari simpul itu sendiri dan juga derajat dari simpul yang terhubung dengannya. Secara intuitif, ukuran ini mempertimbangkan tidak hanya berapa banyak pengguna yang dikenal, tetapi juga siapa yang dikenal. Dengan kata lain, Nilai *eigenvector centrality* dari simpul i sebanding dengan jumlah *eigenvector centrality* dari simpul tetangga. Hal ini adalah asumsi dari formula *eigenvector centrality* pada simpul i sebagai berikut (Oliveira & Gama, 2012).

$$x_i = \frac{1}{\lambda} \sum_{j=1}^n a_{i,j} \cdot x_j \quad (2.12)$$

$$a_{i,j} = \begin{cases} 1, & \text{apabila terdapat sisi di antara simpul } i \text{ dan } j \\ 0, & \text{yang lainnya} \end{cases}$$

keterangan:

- $a_{i,j}$: matriks ketetanggaan
- x_j : nilai sentralitas dari simpul j
- λ : *eigenvalue* terbesar

Dengan beberapa penataan ulang, formula di atas dapat ditulis dalam notasi vektor menjadi persamaan *eigenvector* sebagai berikut (Fornito dkk., 2016).

$$(A - \lambda I)x = 0 \quad (2.13)$$

Untuk menghitung metrik ini, dibutuhkan *eigenvalues* dan *eigenvectors* dari matriks ketetanggaan. Untuk mencari *eigenvalues* maka kita dapat menggunakan persamaan karakteristik polinomial sebagai berikut (Fornito dkk., 2016).

$$|A - \lambda I| = 0 \quad (2.14)$$

keterangan:

- A : matriks ketetanggaan berukuran $n \times n$
- λ : *eigenvalues*
- I : matriks identitas

Selanjutnya, yaitu mencari *eigenvector* dengan menggunakan *eigenvalue* terbesar, untuk langkah-langkah perhitungannya adalah sebagai berikut.

$$A\vec{v} = \lambda\vec{v} \quad (2.15)$$

$$(A - \lambda I)\vec{v} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \quad (2.16)$$

keterangan:

A : matriks ketetanggaan berukuran $n \times n$

λ : *eigenvalue* terbesar

\vec{v} : *eigenvector*

I : matriks identitas

Notasi \vec{v} merupakan *eigenvector* yang dapat direpresentasikan sebagai matriks $n \times 1$ sebagai berikut:

$$\vec{v} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad (2.17)$$

keterangan:

\vec{v} : *eigenvector*

x_i : *eigenvector centrality* simpul i

Nilai metrik *eigenvector centrality* dari simpul i dapat didefinisikan sebagai urutan ke- i di dalam *eigenvector* \vec{v} . Untuk menormalisasi nilai *eigenvector centrality* dari suatu simpul dapat dihitung dengan cara membagi seluruh nilai *eigenvector* \vec{v} dengan nilainya yang tertinggi (Fornito dkk., 2016).

Rangkuman keseluruhan metrik *centrality* yang digunakan pada penelitian ini ditunjukkan pada Tabel 2.5.

Tabel 2.8 Rangkuman Metrik *Centrality*

No.	Centrality	Deskripsi	Perhitungan
1.	<i>Degree</i>	Simpul dengan derajat tertinggi.	$C_D(i) = \sum_{i \neq j} a_{v_i v_j}$
2.	<i>Betweenness</i>	Simpul yang berada di tengah simpul lainnya.	$C_B(i) = \frac{2}{(n-1)(n-2)} \cdot \sum_{h \neq i, h \neq j, j \neq i} \frac{\sigma_{hj}(i)}{\sigma_{hj}}$
3.	<i>Closeness</i>	Simpul yang paling dekat dengan simpul lainnya.	$C_c(i) = \frac{n-1}{\sum_{i \neq j} d_{v_i} d_{v_j}}$
4.	<i>Eigenvector</i>	Simpul dengan koneksi terbaik.	$x_i = \frac{1}{\lambda} \sum_{j=1}^n a_{i,j} \cdot x_j$

BAB III METODOLOGI

3.1. Objek dan Aspek Penelitian

Objek dari penelitian ini adalah kumpulan data *tweets* sebesar 39.219 baris dan 6 kolom yang diperoleh dari Twitter dengan kata kunci “ruangguru” (terdiri dari 5.488 pengguna dan 4.982 percakapan) dan “zenius” (terdiri dari 2.605 pengguna dan 2.123 percakapan) sejak 1 Juli 2021 sampai dengan 30 September 2021. Proses pengumpulan data pada penelitian ini dilakukan dengan metode *scraping* pada situs jejaring *online* Twitter yang dapat diakses bebas oleh publik. Metode atau pendekatan yang digunakan pada penelitian ini adalah analisis jaringan sosial atau *Social Network Analysis* (SNA) terhadap percakapan pengguna Twitter mengenai perusahaan *educational technology*. Pengukuran SNA pada penelitian ini dibagi menjadi dua bagian, yaitu *network properties* dan *centrality*, dengan beberapa metrik pada setiap pengukurannya. Berdasarkan metode yang digunakan, penelitian ini merupakan penelitian dengan metode kuantitatif karena merupakan penelitian yang dilaksanakan melibatkan investigasi sistematis terhadap suatu fenomena dengan mengumpulkan data yang dapat diukur dengan teknik statistika, matematika, atau komputasi. Berdasarkan tujuan, penelitian ini merupakan penelitian deskriptif, karena didesain untuk mendeskripsikan karakteristik dari objek penelitian.

3.2. Peralatan Penunjang Penelitian

Penelitian ini menggunakan beberapa perangkat lunak untuk menunjang proses pengerjaan dan analisis pada penelitian. Penjelasan dan spesifikasi perangkat lunak yang digunakan pada penelitian ini ditunjukkan sebagai berikut. Bahasa pemrograman yang digunakan pada penelitian ini adalah Python. Python digunakan pada proses *scraping* data, membersihkan data, melakukan transformasi data, membuat model jaringan, dan membuat visualisasi data. Bahasa pemrograman ini juga menyediakan berbagai fungsi yang diperlukan untuk menangani persoalan statistika dan teori graf, dengan bantuan *package* atau *library* yang telah disediakan. Versi bahasa pemrograman Python yang digunakan pada penelitian ini adalah 3.8.10.

Twint adalah suatu *library* dari bahasa pemrograman Python yang mempunyai fungsi sebagai *web scraper* untuk mengunduh konten atau mengekstrak kumpulan data *tweet* dari media sosial Twitter. Data *tweet* yang diunduh oleh Twint adalah data yang bersifat publik dari Twitter. Twint digunakan pada tahapan *data collection* pada penelitian ini.

Pandas adalah suatu *library* dari bahasa pemrograman Python yang menyediakan struktur data dan analisis data yang diperlukan untuk membersihkan data mentah ke dalam sebuah bentuk yang bisa diolah. Salah satu struktur data dasar pada Pandas adalah *dataframe*, yang berguna untuk memudahkan dalam membaca suatu *file* dengan format CSV dan menjadikannya ke dalam bentuk tabel. Pandas digunakan pada tahapan *data pre-processing* pada penelitian ini.

Networkx merupakan *library* dari Python yang berfungsi untuk pembuatan, pengolahan, dan studi tentang struktur jaringan. NetworkX menyediakan struktur data untuk graf dan berbagai algoritma graf untuk menghitung dan menganalisis struktur dari jaringan. NetworkX digunakan pada tahapan SNA pada penelitian ini.

Gephi adalah perangkat lunak *open-source* yang dapat digunakan untuk melakukan visualisasi, menganalisis, dan mengolah jaringan yang berukuran besar. Gephi digunakan pada tahapan visualisasi model jaringan pada penelitian ini.

Jupyter Notebook adalah aplikasi *server-client* yang memungkinkan untuk membuat dan menjalankan dokumen *notebook* melalui *web browser*. Jupyter Notebook digunakan sebagai *code editor* untuk bahasa pemrograman Python pada keseluruhan penelitian ini.

3.3. Tahapan Pelaksanaan Penelitian

Langkah-langkah sistematis yang dilakukan dalam proses pengerjaan pada penelitian ini ditunjukkan pada langkah-langkah sebagai berikut:

1. Studi Literatur

Pada tahapan ini, penulis melakukan studi literatur yang mendukung topik penelitian. Literatur dapat berupa buku, jurnal penelitian sebelumnya, maupun artikel dari internet yang berkaitan dengan topik penelitian yang dilakukan oleh penulis.

2. Identifikasi Permasalahan

Identifikasi permasalahan adalah tahapan yang dilakukan untuk merumuskan permasalahan yang akan diteliti oleh penulis. Rumusan masalah pada penelitian ini yaitu bagaimana penerapan SNA dalam menganalisis dan membandingkan struktur jaringan sosial, serta mengidentifikasi *key actor* pada jaringan sosial yang terbentuk berdasarkan percakapan pengguna Twitter mengenai Ruangguru dan Zenius.

3. *Data Collection*

Pengumpulan data dilakukan dengan metode *scraping* dari media sosial Twitter. Proses *scraping* dilakukan dengan menggunakan bahasa pemrograman Python dan *library* Twint. Data yang dikumpulkan merupakan data yang dapat diakses bebas oleh publik.

4. *Data Pre-processing*

Seluruh data yang sudah terkumpul, kemudian dilakukan tahap *data cleaning* untuk menghilangkan data yang tidak relevan agar lebih mudah untuk selanjutnya dilakukan proses analisis, lalu, dilakukan tahap *data transformation*. Tahapan ini dilakukan dengan bahasa pemrograman Python dan *library* Pandas.

5. Analisis *Network Properties*

Data kemudian diproses menggunakan *library* Networkx dari bahasa pemrograman Python untuk menghitung dan membandingkan nilai metrik *network properties* yang dimiliki kedua jaringan. Beberapa metrik *network properties* yaitu *size*, *order*, *density*, *modularity*, *diameter*, *average path length*, *average degree*, dan *connected component*.

6. Analisis *Centrality*

Tahapan analisis *centrality* dilakukan dengan menggunakan *library* Networkx dari bahasa pemrograman Python untuk melihat nilai dari masing-masing metrik *centrality* pada setiap aktor di kedua jaringan. Perhitungan metrik *centrality* yang dilakukan yaitu: *degree centrality*, *betweenness centrality*, *closeness centrality*, dan *eigenvector centrality*.

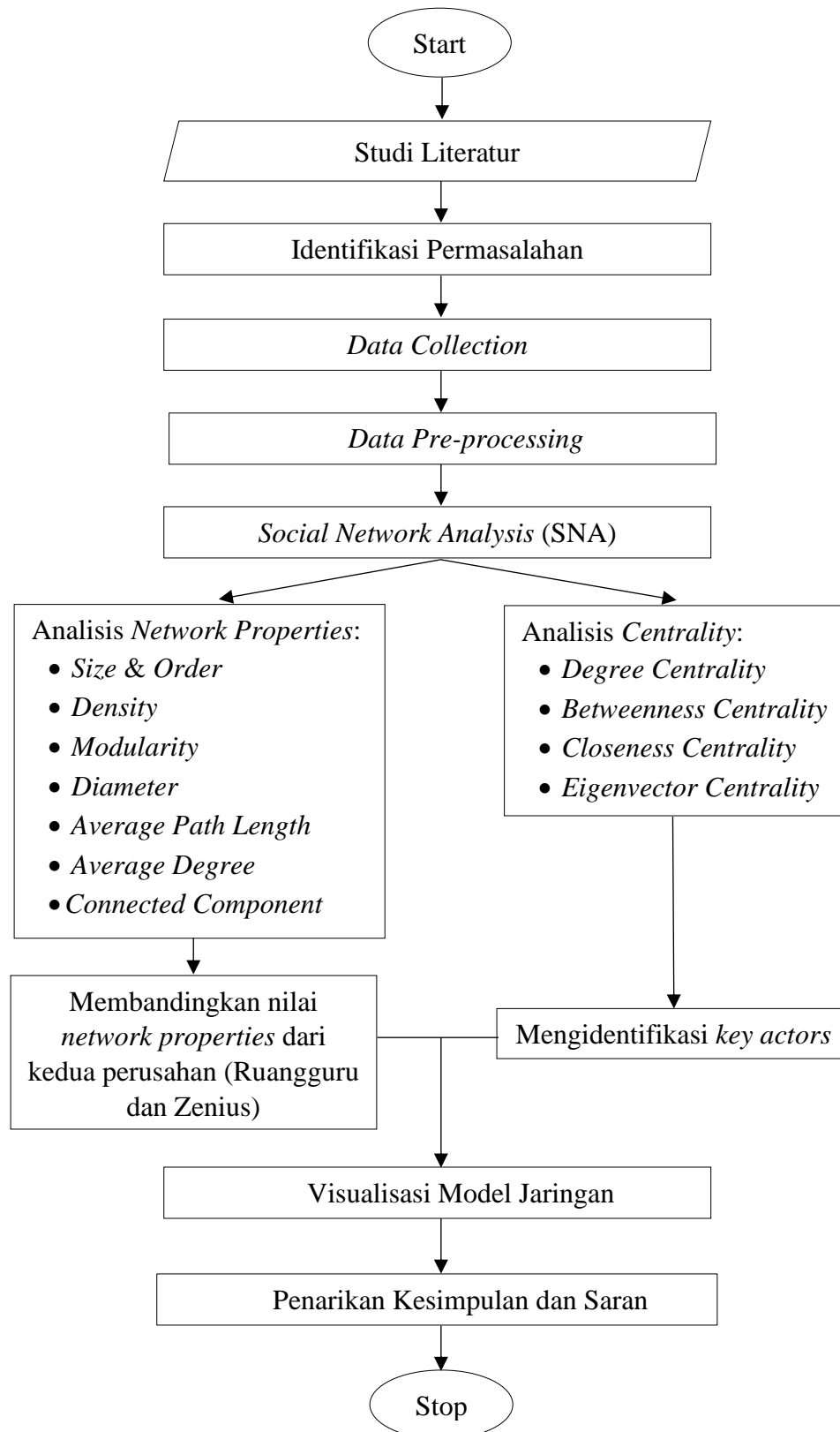
7. Visualiasi Model Jaringan

Data kemudian diproses menggunakan aplikasi Gephi untuk dibuat visualisasi model jaringannya berdasarkan metrik-metrik yang sudah dihitung pada tahapan sebelumnya.

8. Kesimpulan dan Saran

Pada tahapan ini, penulis melakukan penarikan kesimpulan berdasarkan hasil yang telah didapatkan dari tahapan-tahapan sebelumnya. Kemudian, memberikan bahan pertimbangan atau referensi kepada kedua perusahaan tersebut dalam hal pemasaran di media sosial Twitter, dan juga kepada peneliti selanjutnya untuk dapat mengembangkan penelitian yang telah dilakukan oleh penulis.

Langkah-langkah penelitian juga dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian

BAB IV

HASIL DAN PEMBAHASAN

4.1. Data Collection

Tahapan awal pada penelitian ini dimulai dengan mengumpulkan seluruh data yang dibutuhkan dari media sosial Twitter untuk tahapan pengolahan data selanjutnya. Pengambilan data dilakukan dengan metode *scraping*. *Scraping* atau *web scraping* merupakan teknik pengambilan atau ekstraksi data dari suatu *website*, lalu data tersebut umumnya disimpan dalam format *file* tertentu, pada penelitian ini data disimpan ke dalam format *file* CSV. Dalam melakukan *scraping*, dibutuhkan *web scraper* yang merupakan program yang dirancang untuk masuk ke halaman *website* tertentu, lalu mengunduh konten, mengekstrak data, atau hal lain dari *website* tersebut. Pada penelitian ini, proses *web scraping* dilakukan dengan menggunakan suatu *library* dari bahasa pemrograman Python yang bernama Twint.

```
nest_asyncio.apply()
tw = twint.Config()
tw.Search = "ruangguru OR zenius OR zeniuseducation"
tw.Since = "2021-07-01"
tw.Until = "2021-10-30"
tw.Pandas= True
twint.run.Search(tw)
dataset_df = twint.storage.panda.Tweets_df
dataset_df = dataset_df[['id', 'tweet', 'date',
                        'user_id', 'username', 'reply_to']]
dataset_df.to_csv("dataset.csv", index=True, header=True)
```

Kode Program 4.1 *Scraping* Data Twitter

Kode Program 4.1 menunjukkan *script* Python yang digunakan untuk melakukan pengambilan data *tweet* dari Twitter dengan kata kunci pencarian “ruangguru” dan “zenius” pada rentang waktu 1 Juli 2021 sampai dengan 30 September 2021. Tahapan ini menghasilkan suatu *output* berupa *file* dengan format CSV yang berisi data *tweet* hasil *scraping*. Tahapan ini dilakukan berulang kali hingga hasil yang didapatkan sesuai dengan rentang waktu yang telah diinisiasi di awal. Ukuran data mentah yang digunakan pada penelitian ini adalah 39.219 baris dan 5 kolom atribut sebagai berikut.

Tabel 4.1 Deskripsi Atribut pada Data

No.	Atribut	Deskripsi
1.	<i>Id</i>	ID dari <i>tweet</i>
2.	<i>Tweet</i>	isi di dalam pesan (<i>tweet</i>)
3.	<i>Username</i>	nama dari pengguna yang mengirim <i>tweet</i>
4.	<i>User_id</i>	ID dari pengguna yang mengirim <i>tweet</i>
5.	<i>Reply_To</i>	pengguna yang menanggapi <i>tweet</i>

4.2. Data Pre-processing

Setelah mendapatkan data mentah, maka tahapan selanjutnya yang harus dilakukan adalah *data pre-processing* yang bertujuan untuk mengubah data mentah yang telah dikumpulkan menjadi data yang lebih bersih dan bisa digunakan untuk tahap pengolahan atau analisis selanjutnya. Tahapan pada *data pre-processing* dalam penelitian ini adalah sebagai berikut.

4.2.1 Import Dataset

Data yang sebelumnya disimpan dalam bentuk *file* dengan format CSV akan dimasukkan ke dalam *workspace* pada *code editor* dengan menggunakan *library* Pandas dan bahasa pemrograman Python.

```
import pandas as pd
df = pd.read_csv("dataset.csv")
```

Kode Program 4.2 Import Dataset

Kode Program 4.2 adalah potongan *script* Python untuk membaca dan memuat data dari *file* dengan format CSV ke dalam bentuk *dataframe* atau tabel dengan bantuan *library* Pandas. Pandas adalah suatu *library* dari bahasa pemrograman Python yang menyediakan struktur data dan analisis data yang diperlukan untuk membersihkan data mentah ke dalam sebuah bentuk yang bisa diolah. Salah satu struktur data dasar pada Pandas adalah *dataframe*, yang berguna untuk memudahkan dalam membaca suatu *file* dengan format CSV dan menjadikannya ke dalam bentuk tabel.

4.2.2 Menghapus Tweet yang Duplikat

Data *tweet* yang telah dikumpulkan pada tahapan *scraping* memungkinkan masih terdapat data yang duplikat. Oleh karena itu, data duplikat tersebut harus dihapus salah satunya hingga setiap data yang akan dianalisis merupakan data yang unik.

```
df = df.drop_duplicates('id')
```

Kode Program 4.3 Menghapus Tweet yang Duplikat

Kode Program 4.3 adalah *script* Python untuk menghapus salah satu dari data yang duplikat berdasarkan atribut *id* dari *dataframe*. Untuk mengetahui keunikan dari setiap *tweet*, penulis menggunakan atribut *id* yang merupakan suatu representasi dalam *integer* atau bilangan bulat sebagai tanda pengenal unik dari suatu *tweet*.

4.2.3 Menghapus Tweet yang Tidak Mempunyai Interaksi

Berdasarkan batasan masalah yang telah ditetapkan di awal, salah satu batasan pada objek di penelitian ini adalah *tweet* yang mengandung interaksi percakapan atau *reply*. Sehingga, *tweet* yang tidak mempunyai interaksi percakapan ke pengguna lain akan dihapus dari *dataframe*.

```
df = df[df.reply_to != "[]"]
```

Kode Program 4.4 Menghapus Tweet yang Tidak Mempunyai Interaksi

Kode Program 4.4 adalah *script* Python untuk menghapus *tweet* yang tidak mempunyai interaksi percakapan berdasarkan atribut *reply_to* dari *dataframe*. Penulis menggunakan atribut *reply_to* yang merupakan *list* berisi daftar pengguna Twitter yang dilakukan interaksi. Sehingga, jika daftar tersebut tidak mempunyai elemen atau kosong, maka *tweet* tidak mempunyai interaksi. Tahapan ini dilakukan karena akan memudahkan tahapan analisis selanjutnya.

4.2.4 Mengambil Tweet Percakapan Antara Pengguna

Berdasarkan latar belakang yang telah ditetapkan di awal, salah satu hal yang melatarbelakangi penelitian ini adalah menganalisis pola interaksi pengguna di Twitter. Sehingga, penulis hanya akan mengambil *tweet* yang merupakan percakapan antara pengguna dan menghapus *tweet* yang berkaitan langsung dengan akun resmi perusahaan.

```
# make reply_to (username and id) columns
df["reply_to_username"] =
    [eval(df.reply_to.tolist()[i])[0]["screen_name"]
     for i in range(0, df.shape[0])]
df["reply_to_id"] = [eval(df.reply_to.tolist()[i])[0]["id"]
                    for i in range(0, df.shape[0])]
# take only user interaction
df = df[~df.username.str.contains("zeniuseducation|ruangguru")]
df = df[~df.reply_to_username.str.contains("zeniuseducation|ruangguru")]
```

Kode Program 4.5 Mengambil *Tweet* Percakapan Antara Pengguna

Kode Program 4.5 adalah *script* Python untuk menghapus *tweet* yang berkaitan langsung dengan akun resmi perusahaan Ruangguru dan Zenius. Penulis menggunakan atribut *reply_to* dari *dataframe*. Pada atribut ini terdapat informasi mengenai *username* dan *id* akun yang berinteraksi dengan *tweet*. Tahapan ini dilakukan karena *tweet* yang akan dianalisis adalah interaksi percakapan hanya antara pengguna Twitter.

4.2.5 Pengelompokan *Tweet* Mengenai Perusahaan *Edtech*

Berdasarkan batasan masalah yang telah ditetapkan di awal, salah satu batasan pada objek di penelitian ini adalah *tweet* yang mengandung percakapan mengenai Ruangguru atau Zenius. Sehingga, *dataframe* dikelompokkan terlebih dahulu berdasarkan konteks percakapan mengenai Ruangguru atau Zenius.

```
df_zenius = df[
    (df.tweet.str.contains("zenius") == True)
    &
    ~df.tweet.str.contains("ruangguru|rg|ruang guru")].copy()
df_ruangguru = df[
    df.tweet.str.contains("ruangguru|rg|ruang guru") == True
    &
    ~df.tweet.str.contains("zenius")].copy()
```

Kode Program 4.6 Pengelompokan *Tweet* Mengenai Kedua Perusahaan *Edtech*

Kode Program 4.6 adalah *script* Python untuk mengelompokkan *tweet* berdasarkan konteks percakapan mengenai Ruangguru atau Zenius. Untuk mengetahui apakah suatu *tweet* mengandung kata tertentu, maka dapat dilihat dari atribut *tweet* pada *dataframe*. Tahapan ini menghasilkan *output* berupa dua *dataframes*, yaitu untuk Ruangguru dan Zenius. Tahapan ini dilakukan karena analisis jaringan sosial akan dilakukan pada masing-masing jaringan sosial yang terbentuk pada Ruangguru dan Zenius.

4.2.6 Transformasi Data ke Bentuk *Edge List*

Setelah melakukan langkah-langkah *data cleaning*, langkah terakhir pada tahapan *data pre-processing* adalah melakukan *data transformation* atau transformasi pada kedua *dataframes* ke bentuk *edge lists*. *Edge list* merupakan suatu representasi sederhana dari suatu graf. Untuk membentuk suatu *edge list* diperlukan setidaknya dua simpul yang merepresentasikan nama akun yang membalas suatu *tweet* dan nama akun yang dibalas.

```

# function to transform df to edge list form
def transform_edglst(df):
    # source and target column
    edglst = df[['username', 'reply_to_username']].copy().rename({
        "username": "Source",
        "reply_to_username": "Target"},
        # to lower case
        axis=1).applymap(lambda s:s.lower())
    # number of interactions column
    edglst = edglst.value_counts().to_frame('Jumlah Interaksi').astype(
        float)
    return edglst.reset_index()
# apply the function
[edglst_ruangguru, edglst_zenius] = [transform_edglst(x)
                                     for x in [df_ruangguru, df_zenius]]

```

Kode Program 4.7 Transformasi Data ke Bentuk *Edge List*

Kode Program 4.7 adalah *script* Python untuk melakukan transformasi pada *dataframe* ke bentuk *edge list*. Pertama, penulis membuat atribut *Source* dan *Target* pada *edge list*. Atribut *Source* berisi kumpulan *nodes* yang merepresentasikan nama akun yang membalas *tweet*, sedangkan atribut *Target* berisi kumpulan simpul yang merepresentasikan nama akun yang dibalas. Selanjutnya, seluruh data yang bertipe *string* akan dikonversi menjadi *lower case*, hal ini dilakukan untuk memudahkan analisis kedepan karena seluruh data sudah konsisten sepenuhnya dalam huruf kecil. Lalu, penulis juga membuat atribut Jumlah Interaksi yang merupakan banyaknya interaksi antara simpul *source* dan *target*. Ukuran dari kedua data *edge list* yang dihasilkan pada tahapan ini adalah 5.231 baris dan 3 kolom untuk Ruangguru dan 2.156 baris dan 3 kolom untuk Zenius.

4.3. Social Network Analysis (SNA)

Tahapan selanjutnya adalah mengolah data *edge list* Ruangguru dan Zenius menggunakan pendekatan SNA. Pertama, penulis akan membuat dua graf Networkx berdasarkan kedua data *edge list* yang telah kita miliki.

```

g_ruangguru , g_zenius =
[nx.from_pandas_edgelist(x, source='Source', target='Target')
 for x in [edglst_ruangguru, edglst_zenius]]
# convert undirected graph
g_ruangguru = g_ruangguru.to_undirected()
g_zenius = g_zenius.to_undirected()

```

Kode Program 4.8 Transformasi *Edge List* ke Graf Tanpa Arah

Kode Program 4.8 adalah *script* Python untuk melakukan transformasi *edge list* ke bentuk graf tanpa arah. Graf tanpa arah dipilih karena pada penelitian ini penulis hanya berfokus untuk menganalisis persebaran informasi berdasarkan ada atau tidaknya interaksi percakapan yang dapat dinotasikan sebagai sisi pada sepasang simpul di jaringan. Sehingga, arah interaksi atau urutan simpul tidak termasuk dalam fokus penelitian ini. Networkx merupakan *library* dari Python yang berfungsi untuk pembuatan, pengolahan, dan studi tentang struktur jaringan. Dalam penelitian ini, penulis membagi pengukuran pada SNA menjadi dua bagian, yaitu *network properties* dan *centrality*.

4.3.1 Network Properties

Metrik *network properties* berguna untuk mendeskripsikan struktur suatu model jaringan. Adapun beberapa metrik *network properties* pada penelitian ini antara lain: *order*, *size*, *density*, *modularity*, *diameter*, *average path length*, *average degree*, dan *connected component*.

4.3.1.1 Size dan Order

Untuk menghitung nilai *order* (banyaknya simpul) dan nilai *size* (banyaknya sisi) pada jaringan, penulis menggunakan fungsi *order* dan *size* dari library Networkx.

```
size_ruangguru, order_ruangguru = [g_ruangguru.size(), g_ruangguru.order()]
size_zenius, order_zenius = [g_zenius.size(), g_zenius.order()]
```

Kode Program 4.9 Order dan Size pada Jaringan

Kode Program 4.9 adalah *script* Python untuk menghitung nilai *order* dan *size* pada masing-masing jaringan. Nilai *order* dan *size* pada jaringan Ruangguru masing-masing adalah 4.982 dan 5.488, sedangkan untuk jaringan Zenius adalah 2.123 dan 2.605. Pada metrik ini, Ruangguru unggul dibandingkan Zenius. Hal ini menunjukkan bahwa secara relatif lebih banyak pengguna Twitter yang melakukan percakapan mengenai Ruangguru dibandingkan Zenius.

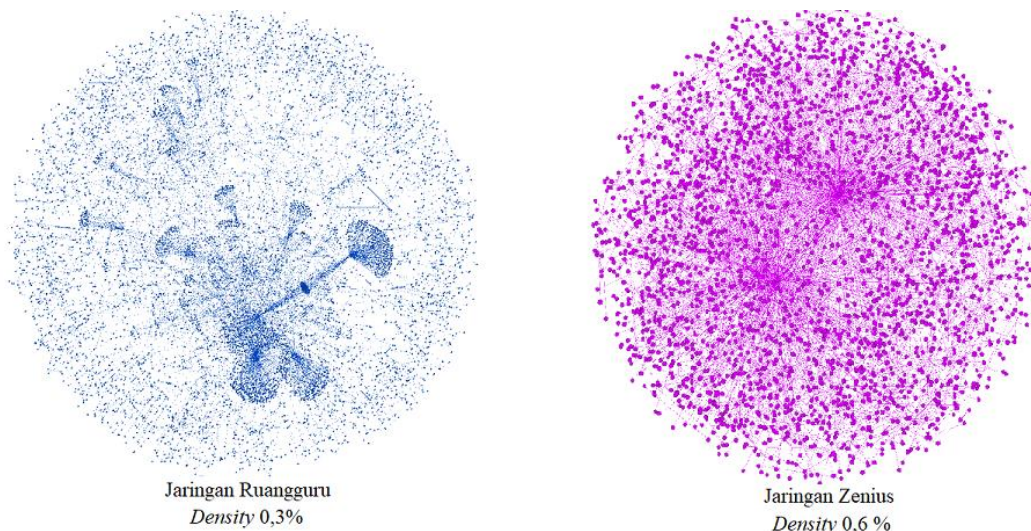
4.3.1.2 Density

Untuk menghitung nilai *density* atau kerapatan pada jaringan, penulis menggunakan fungsi *density* dari library Networkx.

```
density_ruangguru = nx.classes.function.density(g_ruangguru)
density_zenius = nx.classes.function.density(g_zenius)
```

Kode Program 4.10 Density pada Jaringan

Kode Program 4.10 adalah *script* Python untuk menghitung nilai *density* pada masing-masing jaringan. Nilai *density* untuk jaringan Ruangguru adalah 0,00033089 (0,3%) sedangkan nilai *density* untuk jaringan Zenius adalah 0,00062594 (0,6%). Pada metrik ini, Zenius unggul dibandingkan Ruangguru. Hal ini menunjukkan bahwa pengguna Twitter pada jaringan percakapan mengenai Zenius secara relatif lebih saling terhubung satu sama lain dibandingkan pada jaringan percakapan mengenai Ruangguru.



Gambar 4.1 Density Jaringan Ruangguru dan Zenius

4.3.1.3 Modularity

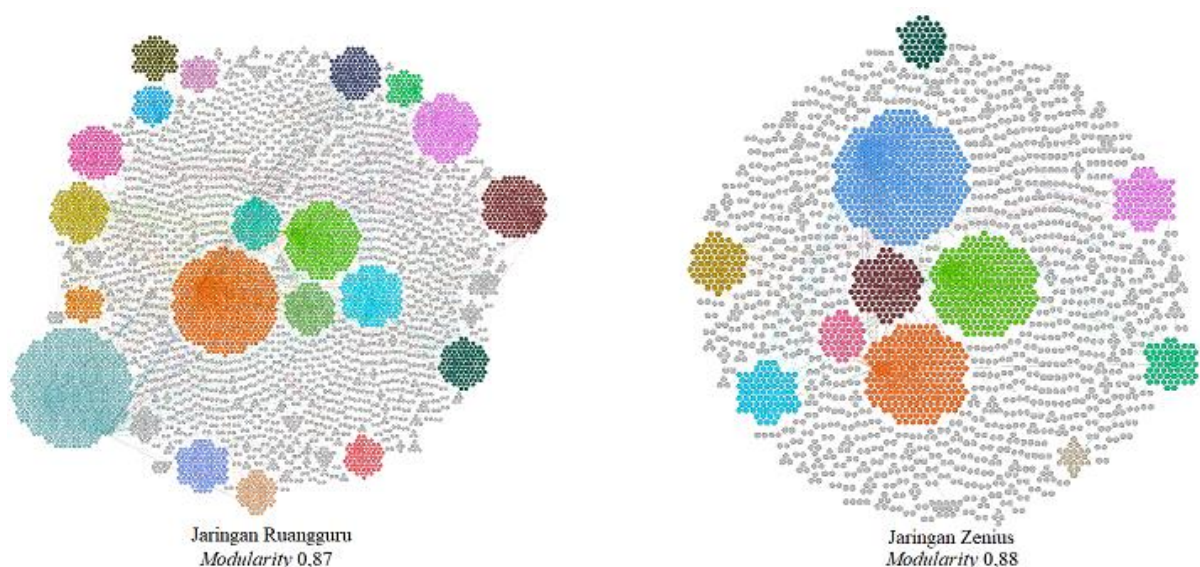
Untuk menghitung nilai *modularity* atau kualitas pembagian jaringan ke dalam kelompok dengan algoritma Louvain, penulis menggunakan *library* Python-Louvain atau Community dari bahasa pemrograman Python. *Library* ini berfungsi sebagai implementasi *community detection* untuk graf Networkx dengan menggunakan algoritma Louvain.

```
import community
# modularity function
def modularity(g):
    mod = community.modularity(community.best_partition(g), g)
    return mod
# implement the function
mod_ruangguru, mod_zenius = [modularity(g) for g in [g_ruangguru,
                                                    g_zenius]]
```

Kode Program 4.11 Modularity pada Jaringan

Kode Program 4.11 merupakan *script* Python untuk menghitung nilai *modularity* pada jaringan dengan menggunakan algoritma Louvain. Nilai *modularity* untuk jaringan Ruangguru adalah 0,873 dan nilai *modularity* untuk jaringan Zenius adalah 0,888. Pada metrik ini, Zenius sedikit lebih unggul dibandingkan Ruangguru. Hal ini menunjukkan bahwa kelompok atau klaster yang terbentuk dalam jaringan percakapan mengenai Zenius memiliki struktur yang secara relatif lebih baik dibandingkan pada jaringan percakapan mengenai Ruangguru. Nilai *modularity* yang lebih baik menunjukkan struktur kelompok yang terbentuk terhubung erat secara internal kelompok dibandingkan terhubung secara eksternal.

Setelah melakukan proses perhitungan metrik *modularity*, pada jaringan Ruangguru terbentuk 20 kelompok besar (kelompok yang mempunyai presentase perbandingan jumlah anggota kelompok dengan jumlah aktor diseluruh jaringan lebih besar atau sama dengan 1%), sedangkan pada jaringan Zenius terbentuk 11 kelompok besar. Kelompok-kelompok besar tersebut adalah kelompok yang diberikan warna pada visualisasi jaringan yang ditunjukkan pada Gambar 4.2. Walaupun jaringan Ruangguru memiliki lebih banyak kelompok besar, namun dapat terlihat bahwa pada jaringan Zenius lebih terhubung erat secara internal kelompok dibandingkan terhubung secara eksternal.



Gambar 4.2 Modularity Jaringan Ruangguru dan Zenius

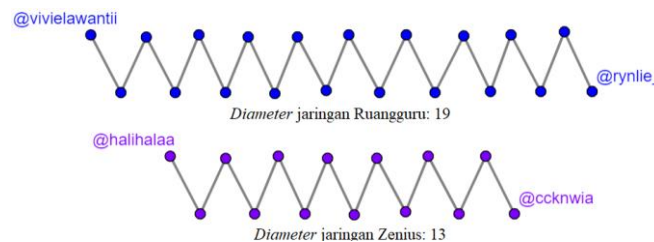
4.3.1.4 Diameter

Untuk menghitung nilai *diameter* atau jarak lintasan terpendek terjauh antara sepasang simpul di dalam jaringan, penulis menggunakan fungsi *all pairs shortest-path length* dari library Networkx dan fungsi *max* dari bahasa pemrograman Python.

```
# diameter function
def diameter(g):
    return max([max(j.values()) for (i,j) in
                nx.all_pairs_shortest_path_length(g)])
# implement the function
d_ruangguru, d_zenius = [diameter(x) for x in [g_ruangguru, g_zenius]]
```

Kode Program 4.12 Diameter pada Jaringan

Kode Program 4.12 adalah *script* Python untuk menghitung nilai *diameter* pada jaringan. Pertama, penulis mencari jarak lintasan terpendek pada seluruh pasang simpul di jaringan dengan fungsi *all pairs shortest-path length* dari library Networkx. Selanjutnya, dari seluruh jarak lintasan terpendek tersebut, dicari jarak yang paling besar menggunakan fungsi *max* dari bahasa pemrograman Python. Nilai *diameter* untuk jaringan Ruangguru adalah 19 dan untuk jaringan Zenius adalah 13. Pada metrik ini, Zenius unggul dibandingkan Ruangguru, karena semakin kecil nilai *diameter* maka semakin baik. Hal ini menunjukkan bahwa proses penyebaran informasi dari suatu aktor kepada aktor yang terjauh di jaringan percakapan mengenai Zenius memakan waktu secara relatif lebih sedikit dibandingkan pada jaringan percakapan mengenai Ruangguru karena hanya melibatkan lebih sedikit aktor.



Gambar 4.3 Diameter pada Jaringan Ruangguru dan Zenius

4.3.1.5 Average Path Length

Karena kedua jaringan memiliki jumlah *connected components* lebih dari satu, maka untuk menghitung nilai *average path length* menggunakan jumlah jarak lintasan terpendek pada setiap simpul di jaringan dibagi dengan banyaknya lintasan terpendek pada setiap simpul di jaringan. Penulis menggunakan fungsi *shortest-path length* dari library Networkx.

```
# average path length function
def avg_path(g):
    path_lengths = [sum(j.values())
                    for (i,j) in nx.shortest_path_length(g)]
    number_of_paths = [len(j.values())
                      for (i,j) in nx.shortest_path_length(g)]
    result = sum(path_lengths)/sum(number_of_paths)
    return result
# implement the function
avgpath_ruangguru, avgpath_zenius = [avg_path(x) for x in [g_ruangguru,
                                                            g_zenius]]
```

Kode Program 4.13 Average Path Length pada Jaringan

Kode Program 4.13 adalah *script* Python untuk menghitung nilai *average path length* pada jaringan. Nilai *average path length* untuk jaringan Ruangguru adalah 5,2505 dan untuk jaringan Zenius adalah 3,9469. Pada metrik ini, Zenius unggul dibandingkan Ruangguru, karena semakin kecil nilai *average path length* maka semakin baik. Hal ini menunjukkan bahwa rata-rata jarak yang harus ditempuh untuk melakukan penyebaran informasi pada jaringan percakapan mengenai Zenius secara relatif lebih pendek dibandingkan pada jaringan percakapan mengenai Ruangguru.

4.3.1.6 Average Degree

Untuk menghitung nilai *average degree* atau rata-rata dari jumlah sisi yang menghubungkan suatu simpul ke simpul lain di jaringan, penulis menggunakan fungsi *degree* dan *order* dari *library* Networkx dan fungsi *sum* dari bahasa pemrograman Python.

```
# average degree function
def avg_deg(g):
    return sum([val for (node, val) in g.degree()])/g.order()
# implement the function
avgdeg_ruangguru, avgdeg_zenius = [avg_deg(g) for g in [g_ruangguru,
                                                         g_zenius]]
```

Kode Program 4.14 Average Degree pada Jaringan

Kode Program 4.14 adalah *script* Python untuk menghitung nilai *average degree* pada jaringan. Nilai *average degree* untuk jaringan Ruangguru adalah 1,8156 dan untuk jaringan Zenius adalah 1,6299. Pada metrik ini, Ruangguru unggul dibandingkan dengan Zenius. Hal ini menunjukkan bahwa kecepatan penyebaran informasi pada jaringan percakapan mengenai Ruangguru secara relatif lebih cepat dibandingkan pada jaringan percakapan mengenai Zenius, karena satu simpul dapat menyebarkan informasi menuju lebih banyak simpul lainnya.

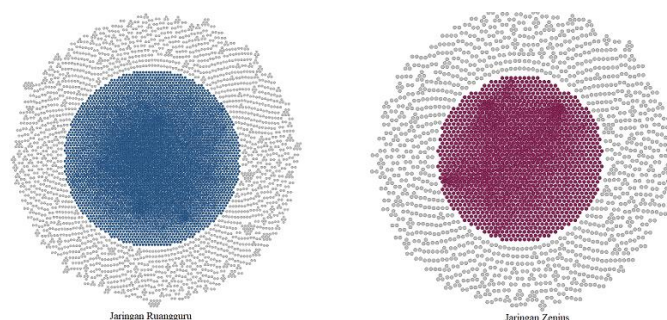
4.3.1.7 Connected Components

Untuk menghitung nilai *connected components* atau kelompok simpul yang tidak terkoneksi dengan keseluruhan jaringan, penulis menggunakan fungsi *number connected components* dari *library* Networkx.

```
conn_ruangguru = nx.number_connected_components(g_ruangguru)
conn_zenius = nx.number_connected_components(g_zenius)
```

Kode Program 4.15 Connected Components pada Jaringan

Kode Program 4.15 adalah *script* Python untuk menghitung nilai *connected components* pada jaringan. Nilai *connected components* dari jaringan Ruangguru adalah 1.022 dan untuk jaringan Zenius adalah 587. Pada metrik ini, Zenius unggul dibandingkan Ruangguru karena semakin kecil nilai *connected components* maka semakin baik. Hal ini menunjukkan bahwa pada jaringan percakapan mengenai Zenius, simpul secara relatif tidak terlalu banyak terpisah ke dalam kelompok-kelompok kecil yang tidak terkoneksi.



Gambar 4.4 Komponen Utama pada Jaringan Ruangguru dan Zenius

Seluruh nilai metrik *network properties* pada jaringan percakapan pengguna Twitter mengenai Ruangguru dan Zenius ditunjukkan pada Tabel 4.8.

Tabel 4.2 Perbandingan Metrik *Network Properties* pada Jaringan Ruangguru dan Zenius

No.	<i>Network Properties</i>	Ruangguru	Zenius	Deskripsi
1.	<i>Size</i>	4.982	2.123	Jaringan Ruangguru memiliki interaksi yang lebih banyak.
2.	<i>Order</i>	5.488	2.605	Jaringan Ruangguru memiliki aktor yang lebih banyak.
3.	<i>Density</i>	0,3 %	0,6 %	Pada jaringan Zenius, aktor lebih saling terhubung satu sama lain.
4.	<i>Modularity</i>	0,87334	0,88822	Pada jaringan Zenius, kelompok yang terbentuk memiliki struktur yang lebih baik.
5.	<i>Diameter</i>	19	13	Pada jaringan Zenius, proses penyebaran informasi antara suatu aktor dengan aktor yang terjauh hanya perlu melewati sedikit aktor.
6.	<i>Average Path Length</i>	5,2505	3,9469	Pada jaringan Zenius, jarak yang harus ditempuh aktor untuk melakukan penyebaran informasi lebih pendek.
7.	<i>Average Degree</i>	1,8156	1,6299	Pada jaringan Ruangguru, aktor dapat menyebarkan informasi ke lebih banyak aktor lainnya.
8.	<i>Connected Components</i>	1.022	587	Pada jaringan Zenius, aktor tidak banyak terpisah ke kelompok yang tidak terkoneksi.

Tabel 4.8 menunjukkan perbandingan seluruh hasil perhitungan metrik *network properties* pada jaringan Ruangguru dan Zenius. Terlihat bahwa karakteristik struktur jaringan Ruangguru unggul dalam tiga metrik, yaitu *size*, *order*, dan *average degree*. Sedangkan, karakteristik struktur jaringan Zenius unggul dalam lima metrik, yaitu *density*, *modularity*, *diameter*, *average path length*, dan *connected components*.

Berdasarkan hasil perhitungan seluruh metrik *network properties*, dapat terlihat bahwa pada jaringan Ruangguru unggul pada pengukuran dengan tipe *connection*, namun tidak unggul pada pengukuran dengan tipe *distribution* dan *segmentation*. Hal ini mempunyai makna bahwa pada jaringan Ruangguru, walaupun memiliki banyak aktor dan interaksi percakapan, persebaran informasi dan intensitas interaksi percakapan antara banyak aktor-aktor tidak sebaik atau lebih efisien dibandingkan pada jaringan Zenius.

4.3.2 Centrality

Setelah pengukuran *network properties*, pengukuran selanjutnya adalah *centrality*. Pengukuran pada *centrality* bertujuan untuk mengidentifikasi aktor yang paling berpengaruh dalam penyebaran informasi (*key actor*) di antara semua aktor yang ada di dalam suatu jaringan. Terdapat empat pengukuran *centrality* pada penelitian ini, yaitu *degree centrality*, *betweenness centrality*, *closeness centrality*, dan *eigenvector centrality*.

4.3.2.1 Degree Centrality

Untuk menghitung nilai *degree centrality* atau jumlah sisi yang dimiliki suatu simpul di dalam jaringan, penulis menggunakan fungsi *degree centrality* dari library Networkx.

```
# degree centrality function
def degree_c(g):
    res = nx.algorithms.centrality.degree_centrality(g)
    return pd.DataFrame(list(zip(list(res.keys()),
                                  list(res.values()))),
                        columns=['Node',
                                'Degree Centrality']
                        ).sort_values("Degree Centrality",
                                      ascending=False
                                      ).reset_index(drop=True)

# implement the function
df_degreecent_ruangguru, df_degreecent_zenius = [degree_c(x)
                                                  for x in [g_ruangguru,
                                                          g_zenius]]
```

Kode Program 4.16 Degree Centrality pada Jaringan

Kode Program 4.16 adalah *script* Python untuk menghitung nilai *degree centrality* pada seluruh aktor di masing-masing jaringan. Lima aktor dengan peringkat tertinggi pada jaringan percakapan pengguna Twitter mengenai Ruangguru ditunjukkan pada Tabel 4.9.

Tabel 4.3 Lima Aktor dengan Degree Centrality Tertinggi pada Jaringan Ruangguru

No.	Aktor	Degree Centrality
1.	schfess	0,092218
2.	subschfess	0,063787
3.	ambisfs	0,060324
4.	sbmptnfess	0,045380
5.	guidance204	0,019136

Tabel 4.9 menunjukkan lima aktor teratas pada metrik *degree centrality* untuk jaringan Ruangguru, kelima aktor tersebut adalah akun Schfess, Subschfess, Ambisfs, Sbmptnfess, dan Guidance204. Selanjutnya, lima aktor dengan peringkat tertinggi pada jaringan percakapan pengguna Twitter mengenai Zenius ditunjukkan pada Tabel 4.10.

Tabel 4.4 Lima Aktor dengan Degree Centrality Tertinggi pada Jaringan Zenius

No.	Aktor	Degree Centrality
1.	schfess	0,14132
2.	sbmptnfess	0,079109
3.	subschfess	0,071044
4.	sabdaps	0,018817
5.	zenius_oliv	0,018433

Tabel 4.10 menunjukkan lima aktor teratas pada metrik *degree centrality* untuk jaringan Zenius, kelima aktor tersebut adalah akun Schfess, Sbmptnfess, Subschfess, Sabdaps, dan Zenius_oliv. Aktor-aktor ini merupakan aktor yang memiliki *social connections* terbanyak dibandingkan aktor-aktor lain di dalam jaringan.

4.3.2.2 Betweenness Centrality

Untuk menghitung nilai *betweenness centrality* atau proporsi lintasan terpendek di antara semua pasangan simpul di dalam jaringan yang melewati suatu simpul tertentu, penulis menggunakan fungsi *betweenness centrality* dari library Networkx.

```
# betweenness centrality function
def betweenness_c(g):
    res = nx.algorithms.centrality.betweenness_centrality(g, normalized=True)
    return pd.DataFrame(
        list(zip(list(res.keys()),
                  list(res.values()))), columns=['Node', 'Betweenness Centrality']
    ).sort_values("Betweenness Centrality", ascending=False)
    ).reset_index(drop=True)
# implement the function
df_betwenncent_ruangguru, df_betwenncent_zenius = [betweenness_c(x)
                                                    for x in [g_ruangguru,
                                                            g_zenius]]
```

Kode Program 4.17 *Betweenness Centrality* pada Jaringan

Kode Program 4.17 adalah *script* Python untuk menghitung nilai *betweenness centrality* pada seluruh aktor di masing-masing jaringan. Lima aktor dengan peringkat tertinggi pada jaringan percakapan pengguna Twitter mengenai Ruangguru ditunjukkan pada Tabel 4.11 dan untuk Zenius pada Tabel 4.12.

Tabel 4.5 Lima Aktor dengan *Betweenness Centrality* Tertinggi pada Jaringan Ruangguru

No.	Aktor	<i>Betweenness Centrality</i>
1.	schfess	0,13269
2.	subschfess	0,089669
3.	forskyblue_	0,076197
4.	hiromi_daiji	0,069051
5.	sbmptnfess	0,058783

Tabel 4.11 menunjukkan lima aktor teratas pada metrik *betweenness centrality* untuk jaringan Ruangguru, kelima aktor tersebut adalah akun Schfess, Subschfess, Forskyblue, Hiromi_daiji, dan Sbmptnfess. Selanjutnya, lima aktor dengan peringkat tertinggi pada jaringan percakapan pengguna Twitter mengenai Zenius ditunjukkan pada Tabel 4.12.

Tabel 4.6 Lima Aktor dengan *Betweenness Centrality* Tertinggi pada Jaringan Zenius

No.	Aktor	<i>Betweenness Centrality</i>
1.	schfess	0,14266
2.	sbmptnfess	0,076758
3.	subschfess	0,073354
4.	zenius_oliv	0,039600
5.	hopefullyperf	0,030079

Tabel 4.12 menunjukkan lima aktor teratas pada metrik *betweenness centrality* untuk jaringan Ruangguru, kelima aktor tersebut adalah akun Schfess, Sbmptnfess, Subschfes, Zenius_oliv, dan Hopefullyperf. Aktor-aktor ini merupakan aktor yang terletak di antara aktor-aktor lainnya di dalam jaringan.

4.3.2.3 Closeness Centrality

Karena kedua jaringan memiliki jumlah *connected components* lebih dari satu, untuk menghitung nilai *closeness centrality* atau rata-rata pada seluruh lintasan terpendek dari suatu aktor ke setiap aktor lainnya, penulis menggunakan fungsi *closeness centrality* dengan formula yang disempurnakan oleh Wasserman dan Faust dari *library Networkx*.

```
# closeness centrality function
def closeness_c(g):
    res = nx.algorithms.centrality.closeness_centrality(g, wf_improved=True)
    return pd.DataFrame(
        list(zip(list(res.keys()),
                  list(res.values()))), columns=['Node', 'Closeness Centrality']
    ).sort_values("Closeness Centrality", ascending=False).reset_index(drop=True)
# implement the function
df_closecent_ruangguru, df_closecent_zenius = [closeness_c(x)
                                                for x in [g_ruangguru, g_zenius]]
```

Kode Program 4.18 Closeness Centrality pada Jaringan

Kode Program 4.18 adalah *script* Python untuk menghitung nilai *closeness centrality* pada seluruh aktor di masing-masing jaringan. Lima aktor dengan peringkat tertinggi pada jaringan percakapan pengguna Twitter mengenai Ruangguru ditunjukkan pada Tabel 4.13.

Tabel 4.7 Lima Aktor dengan *Closeness Centrality* Tertinggi pada Jaringan Ruangguru

No.	Aktor	Closeness Centrality
1.	schfess	0,18217
2.	hiromi_daiji	0,17051
3.	subschfess	0,17028
4.	syafiranurainun	0,16526
5.	sbmptnfess	0,16497

Tabel 4.13 menunjukkan lima aktor teratas pada metrik *closeness centrality* untuk jaringan Ruangguru, kelima aktor tersebut adalah akun Schfess, Hiromi_daiji, Subschfess, Syafiranurainun, dan Sbmptnfess. Selanjutnya, lima aktor dengan peringkat tertinggi pada jaringan percakapan pengguna Twitter mengenai Zenius ditunjukkan pada Tabel 4.14.

Tabel 4.8 Lima Aktor dengan *Closeness Centrality* Tertinggi pada Jaringan Zenius

No.	Aktor	Closeness Centrality
1.	schfess	0,20610
2.	zenius_oliv	0,19737
3.	hopefullyperf	0,19039
4.	byunpov	0,18547
5.	amsterdamlaf	0,18475

Tabel 4.14 menunjukkan lima aktor teratas pada metrik *closeness centrality* untuk jaringan Zenius, kelima aktor tersebut adalah akun Schfess, Zenius_oliv, Hopefullyperf, Hyunpov, dan Amsterdamlaf. Aktor-aktor ini merupakan aktor yang paling dekat dengan aktor-aktor lainnya di dalam jaringan.

4.3.2.4 Eigenvector Centrality

Untuk menghitung nilai *eigenvector centrality* atau nilai kualitas koneksi dari suatu aktor di jaringan, penulis menggunakan fungsi *eigenvector centrality* dari library Networkx.

```
# eigenvector centrality function
def eigen_c(g):
    res = nx.algorithms.centrality.eigenvector_centrality(g, weight=None)
    return pd.DataFrame(
        list(zip(list(res.keys()),
                list(res.values())
                )),
        columns=['Node',
                'Eigenvector Centrality']
    ).sort_values("Eigenvector Centrality",
                  ascending=False
    ).reset_index(drop=True)

# implement the function
df_eigencent_ruangguru, df_eigencent_zenius = [eigen_c(x)
                                                for x in [g_ruangguru, g_zenius]]
```

Kode Program 4.19 *Eigenvector Centrality* pada Jaringan

Lima peringkat tertinggi pada jaringan percakapan pengguna Twitter mengenai Ruangguru ditunjukkan pada Tabel 4.15.

Tabel 4.9 Lima Aktor dengan *Eigenvector Centrality* Tertinggi pada Jaringan Ruangguru

No.	Aktor	<i>Eigenvector Centrality</i>
1.	schfess	0,61184
2.	subschfess	0,28570
3.	ambisfs	0,13755
4.	sbmptnfess	0,12086
5.	guidance204	0,072467

Tabel 4.15 menunjukkan lima aktor teratas pada metrik *eigenvector centrality* untuk jaringan Ruangguru, kelima aktor tersebut adalah akun Schfess, Subschfess, Ambisfs, Sbmptnfess, dan Guidance204. Untuk jaringan Zenius ditunjukkan pada Tabel 4.16.

Tabel 4.10 Lima Aktor dengan *Eigenvector Centrality* Tertinggi pada Jaringan Zenius

No.	Aktor	<i>Eigenvector Centrality</i>
1.	schfess	0,69410
2.	subschfess	0,10889
3.	sbmptnfess	0,66089
4.	zenius_oliv	0,054261
5.	hopefullyperf	0,051446

Tabel 4.16 menunjukkan lima aktor teratas pada metrik *eigenvector centrality* untuk jaringan Zenius, kelima aktor tersebut adalah akun Schfess, Subschfess, Sbmptnfess, Zenius_oliv, dan Hopefullyperf. Aktor-aktor ini merupakan aktor yang mempunyai koneksi terbaik di jaringan.

Tabel 4.11 Seluruh Hasil Perhitungan Metrik *Centrality* pada Jaringan Ruangguru

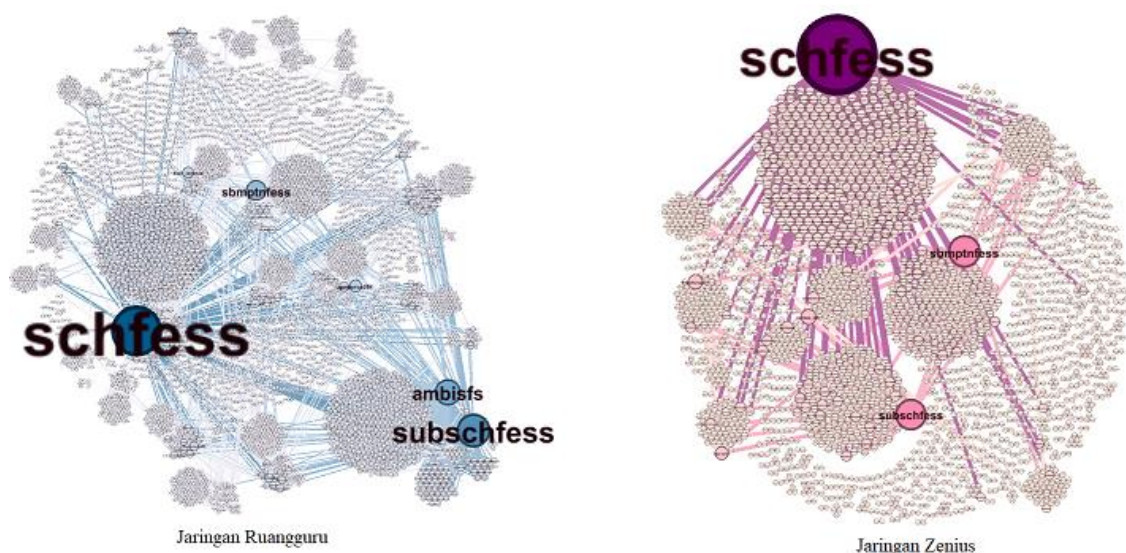
Pengguna	DC	BC	CC	EC
	Score/Rank	Score/Rank	Score/Rank	Score/Rank
schfess	0,0922 / 1	0,132 / 1	0,182 / 1	0,612 / 1
subschfess	0,638 / 2	0,0897 / 2	0,170 / 3	0,286 / 2
ambisfs	0,603 / 3	0,0560 / 6	0,146 / 51	0,137 / 3
sbmptnfess	0,0454 / 4	0,0588 / 5	0,165 / 5	0,121 / 4
guidance204	0,0191 / 5	0,0262 / 9	0,164 / 6	0,0725 / 5

Tabel 4.17 menunjukkan seluruh nilai metrik *centrality* pada jaringan Ruangguru. Dari keempat metrik tersebut, terlihat bahwa aktor dengan nama akun Schfess, Subschfess, dan Sbmptnfess merupakan aktor yang selalu menempati lima besar teratas. Hal ini menunjukkan bahwa ketiga akun tersebut merupakan *key actor* dalam penyebaran informasi pada jaringan sosial percakapan pengguna Twitter mengenai Ruangguru.

Tabel 4.12 Seluruh Hasil Perhitungan Metrik *Centrality* pada Jaringan Zenius

Pengguna	DC	BC	CC	EC
	Score/Rank	Score/Rank	Score/Rank	Score/Rank
schfess	0,141 / 1	0,143 / 1	0,206 / 1	0,694 / 1
sbmptnfess	0,0791 / 2	0,0767 / 2	0,177 / 8	0,0661 / 3
subschfess	0,0710 / 3	0,0733 / 3	0,173 / 9	0,109 / 2
sabdaps	0,0188 / 4	0,0179 / 7	0,132 / 496	0,0024 / 829
zenius_oliv	0,0184 / 5	0,0396 / 4	0,197 / 2	0,0543 / 4

Tabel 4.18 menunjukkan seluruh nilai metrik *centrality* pada jaringan Zenius. Dari keempat metrik tersebut, terlihat bahwa aktor dengan nama akun Schfess merupakan aktor yang selalu menempati lima besar teratas. Hal ini menunjukkan bahwa kedua akun tersebut merupakan *key actor* dalam penyebaran informasi pada jaringan sosial percakapan pengguna Twitter mengenai Zenius.



Gambar 4.5 *Key Actors* pada Jaringan Ruangguru dan Zenius

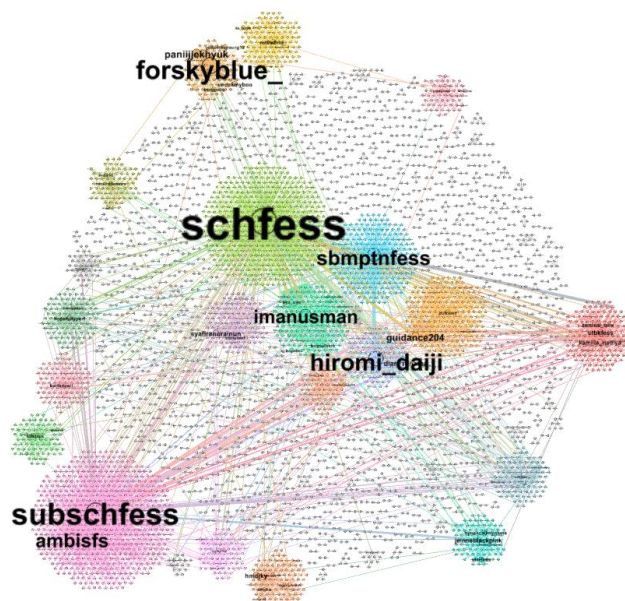
4.4. Visualisasi Model Jaringan

Pada tahapan ini, penulis akan menampilkan representasi visual dari model jaringan yang telah diolah pada tahapan sebelumnya, dengan tujuan agar pembaca lebih mudah untuk mengidentifikasi hasil analisis jaringan. Namun, sebelum itu, penulis akan melakukan tranformasi kedua data graf Networkx ke dalam bentuk *file* Graph Exchange XML Format (GEXF).

```
# adding attributes to graph function
def data_to_gephi(g):
    nx.set_node_attributes(g, community.best_partition(g, resolution=1),
        "Modularity Class")
    nx.set_node_attributes(g, nx.betweenness_centrality(g),
        "Betweenness Centrality")
    return g
# implement the function
g_ruangguru, g_zenius = [data_to_gephi(x) for x in [g_ruangguru, g_zenius]]
nx.write_gexf(g_ruangguru, "ruangguru.gexf")
nx.write_gexf(g_zenius, "zenius.gexf")
```

Kode Program 4.20 Transformasi Data Graf Networkx ke *File* GEXF

Kode Program 4.20 adalah *script* Python untuk melakukan tranformasi data graf Networkx ke *file* dengan format GEXF. GEXF merupakan format yang digunakan untuk menggambarkan struktur jaringan beserta atributnya dan dapat digunakan sebagai format pertukaran antara aplikasi grafik. Terdapat dua atribut pada jaringan yang akan membantu dalam pengolahan visualisasi jaringan pada aplikasi Gephi, yaitu atribut pada metrik *modularity* dan *betweenness centrality*. Metrik *modularity* dibutuhkan pada tahap visualisasi karena dapat menunjukkan kelompok atau kluster yang terbentuk pada jaringan dan metrik *betweenness centrality* dibutuhkan pada tahap visualisasi karena dapat menunjukkan *key actor* yang berperan sentral dalam penyebaran informasi di masing-masing kelompok dan di dalam jaringan. Selanjutnya, kedua files GEXF tersebut diolah dalam aplikasi Gephi. Gambar 4.1 adalah visualisasi jaringan Ruangguru. Terdapat 20 kelompok besar di dalam jaringan (kelompok yang diberikan warna). Kelompok yang besar adalah kelompok yang mempunyai persentase jumlah anggota kelompok dibandingkan dengan jumlah aktor di jaringan di atas atau sama dengan 1 %.



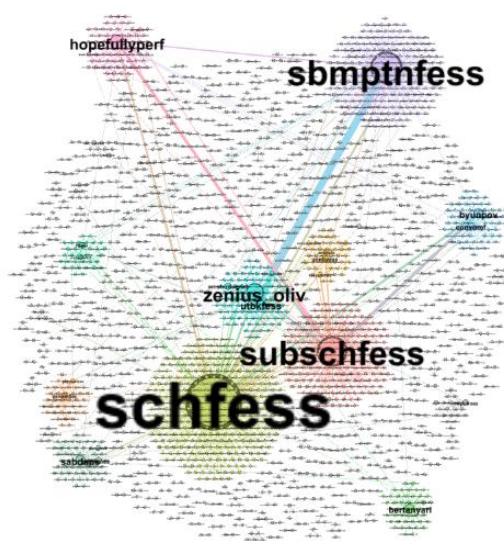
Gambar 4.6 Visualisasi Jaringan Ruangguru

Daftar lima kelompok terbesar pada jaringan Ruangguru beserta atributnya ditunjukkan pada Tabel 4.19.

Tabel 4.13 Lima Kelompok Terbesar pada Jaringan Ruangguru

No.	No. Kelompok	Jumlah Anggota	Persentase Anggota	Key Actor
1.	10	581	10,59 %	subschfess
2.	6	452	8,24 %	schfess
3.	4	230	4,19 %	sbmptnfess
4.	8	204	3,72 %	guidance204
5.	0	165	3,01 %	imanusman

Selanjutnya, untuk visualisasi model jaringan percakapan pengguna Twitter mengenai Zenius ditunjukkan pada Gambar 4.2.



Gambar 4.7 Visualisasi Jaringan Zenius

Gambar 4.2 adalah visualisasi model jaringan percakapan pengguna Twitter mengenai Zenius. Terdapat 11 kelompok atau klaster besar di dalam jaringan (kelompok yang diberikan warna). Daftar lima kelompok terbesar pada jaringan percakapan pengguna Twitter mengenai Zenius beserta atributnya ditunjukkan pada Tabel 4.13.

Tabel 4.14 Lima Kelompok Terbesar pada Jaringan Zenius

No.	No. Kelompok	Jumlah Anggota	Persentase Anggota	Key Actor
1.	5	339	13,01 %	schfess
2.	20	211	8,1 %	sbmptnfess
3.	3	201	7,72 %	subschfess
4.	0	96	3,69 %	zenius_oliv
5.	95	74	2,84 %	hopefullyperf

Berdasarkan data perhitungan seluruh metrik *centrality* pada tahapan sebelumnya dan visualisasi model jaringan di tahapan ini, terlihat bahwa akun yang merupakan *key actor* pada keseluruhan jaringan juga merupakan *key actor* pada kelompok yang besar (kelompok yang mempunyai persentase anggota di atas atau sama dengan 1,00 %) di masing-masing jaringan. Hal ini menunjukkan bahwa selain berpengaruh dalam penyebaran informasi kepada aktor di dalam kelompok masing-masing, *key actors* tersebut juga berpengaruh dalam penyebaran informasi pada seluruh aktor di dalam jaringan.

4.5. Hasil SNA dan Performa Perusahaan Ruangguru dan Zenius

Seluruh perhitungan metrik SNA yang telah dilakukan pada tahapan sebelumnya, pada bagian ini akan dirangkum kembali keseluruhan hasil yang didapatkan. Pertama, pengukuran pada tingkat keseluruhan jaringan, yaitu pengukuran *network properties*. Berdasarkan hasil perhitungan dan perbandingan metrik *network properties* yang telah dilakukan pada tahapan sebelumnya, karakteristik struktur jaringan Ruangguru unggul dalam tiga metrik, yaitu *size*, *order*, dan *average degree*. Sedangkan, karakteristik struktur jaringan Zenius unggul dalam lima metrik, yaitu *density*, *modularity*, *diameter*, *average path length*, dan *connected components*.

Berdasarkan hasil perhitungan seluruh metrik *network properties*, dapat terlihat bahwa pada jaringan Ruangguru unggul pada pengukuran dengan tipe *connection*, namun tidak unggul pada pengukuran dengan tipe *distribution* dan *segmentation*. Hal ini mempunyai makna bahwa pada jaringan Ruangguru, walaupun memiliki banyak aktor dan interaksi percakapan, persebaran informasi dan intensitas interaksi percakapan antara banyak aktor-aktor tidak sebaik atau lebih efisien dibandingkan pada jaringan Zenius.

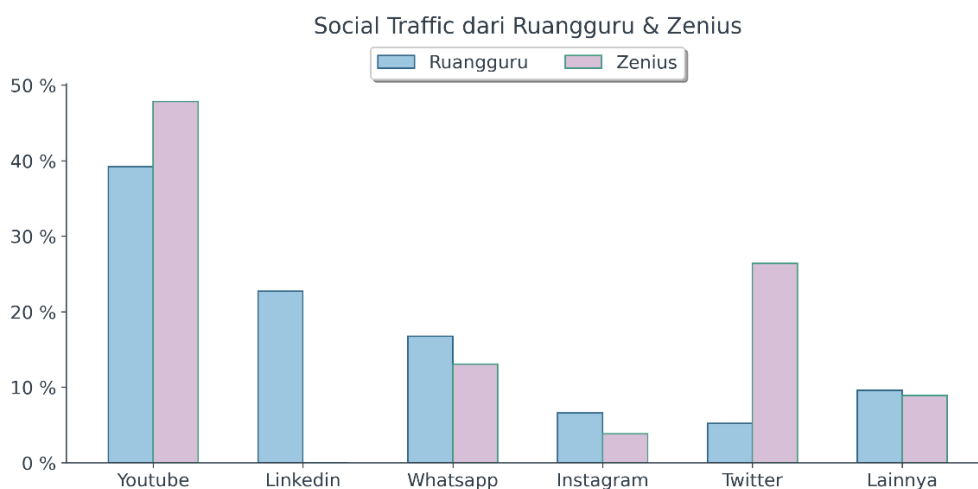
Berdasarkan laporan oleh Similarweb, di mana merupakan platform yang menyediakan analisis *website* perusahaan dan informasi tentang lalu lintas dan kinerja *website* perusahaan pesaing, ditunjukkan *engagement* yang didapat perusahaan Ruangguru dan Zenius dalam jangka waktu Juli 2021 sampai dengan September 2021 pada tabel berikut.

Tabel 4.15 Tingkat *engagement* Perusahaan Ruangguru dan Zenius

No.	Metrik	Ruangguru	Zenius
1.	Rata-rata kunjungan per bulan	19,61 M	2,47 M
2.	Rata-rata pengunjung individu	11,04 M	1,178 M
3.	Rata-rata kunjungan per individu	1,78	2,10
4.	Rata-rata durasi kunjungan	00:02:50	00:06:25
5.	Rata-rata halaman yang dikunjungi	1,67	4,34
6.	Rata-rata pengunjung yang langsung pergi	51,6%	51,39%

Sumber Data: (Similarweb, 2021)

Pada Tabel 4.15 terlihat bahwa, meskipun Ruangguru paling banyak dikunjungi oleh pengguna, namun frekuensi dan durasi waktu kunjungan tidak lebih unggul dibandingkan oleh Zenius. Hal ini dapat menunjukkan bahwa, walaupun Ruangguru dapat menarik lebih banyak pengguna, Zenius dapat membuat penggunaanya lebih sering dan dengan durasi waktu yang lebih lama dalam mengunjungi platform yang mereka miliki.



Gambar 4.8 Social Traffic Perusahaan Ruangguru dan Zenius

Sumber Data: (Similarweb, 2021)

Pada Gambar 4.8 terlihat bahwa *social traffic* yang didapatkan Zenius pada media sosial Twitter dan Youtube lebih unggul dibandingkan dengan Ruangguru. Hal ini menunjukkan bahwa Zenius lebih unggul memanfaatkan kedua media sosial tersebut untuk menarik pengguna agar melihat konten pemasaran, lalu menarik pengguna tersebut untuk mengunjungi platform perusahaan. Hal ini menarik, karena berdasarkan data yang ditunjukkan pada Gambar 1.2, lebih banyak pengguna Twitter yang membicarakan Ruangguru dibandingkan Zenius. Namun setelah perhitungan SNA, struktur jaringan percakapan mengenai Zenius unggul dibandingkan dengan Ruangguru. Hal ini dapat menunjukkan bahwa persebaran informasi dan intensitas percakapan mengenai Zenius lebih unggul dibandingkan mengenai Ruangguru.

Pengukuran selanjutnya adalah pada tingkat unit aktor atau pengukuran untuk mengidentifikasi aktor yang paling berpengaruh dalam penyebaran informasi di jaringan atau biasa disebut *key actor*, pengukuran tersebut adalah yaitu *centrality*. Pada Tabel 4.17 ditunjukkan lima aktor dengan nilai tertinggi pada seluruh nilai metrik *centrality* di jaringan Ruangguru. Dari keempat metrik tersebut, terlihat bahwa aktor dengan nama akun Schfess, Subschfess, dan Sbmptnfess merupakan aktor yang selalu menempati lima besar teratas. Hal ini menunjukkan bahwa ketiga akun tersebut merupakan *key actor* dalam penyebaran informasi pada jaringan sosial percakapan pengguna Twitter mengenai Ruangguru.

Selanjutnya, pada Tabel 4.18 menunjukkan seluruh nilai metrik *centrality* pada jaringan Zenius. Dari keempat metrik tersebut, terlihat bahwa aktor dengan nama akun Schfess dan Zenius_oliv merupakan aktor yang selalu menempati lima besar teratas. Hal ini menunjukkan bahwa kedua akun tersebut merupakan *key actor* dalam penyebaran informasi pada jaringan sosial percakapan pengguna Twitter mengenai Zenius.

Berdasarkan data yang ditunjukkan pada Tabel 2.2 dan Tabel 2.3, terlihat bahwa akun Twitter resmi perusahaan Ruangguru telah bekerja sama dengan akun Amandamanopo, Fikinaki__, Angga_yunandaa, dan Imanusman untuk menjadikan akun-akun tersebut sebagai *brand ambassador* untuk Ruangguru di Twitter. Sedangkan, Zenius telah bekerja sama dengan akun Insisani dan Rsembhada sebagai *brand ambassador* untuk Zenius di Twitter.

Berdasarkan hal-hal yang telah dijelaskan di atas, terlihat bahwa akun Twitter resmi perusahaan Ruangguru dan Zenius belum bekerja sama dengan *key actors* yang mereka miliki di masing-masing jaringan sosial percakapan pengguna Twitter mengenai perusahaan mereka.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan analisis dan hasil penelitian pada bab sebelumnya, dapat disimpulkan:

1. Berdasarkan hasil perhitungan dan perbandingan metrik *network properties*, karakteristik struktur jaringan sosial percakapan pengguna Twitter mengenai Zenius unggul dalam lima dari delapan metrik, yaitu *density*, *modularity*, *diameter*, *average path length*, dan *connected components*. Hal ini menunjukkan bahwa pada jaringan Ruangguru, walaupun memiliki banyak aktor dan interaksi percakapan, persebaran informasi dan intensitas percakapan antara aktor-aktor tersebut tidak lebih baik atau lebih efisien dibandingkan pada jaringan Zenius.
2. Menggunakan perhitungan metrik *degree centrality*, *betweenness centrality*, *closeness centrality*, dan *eigenvector centrality*, didapatkan *key actors* pada masing-masing jaringan. *Key actors* pada jaringan Ruangguru adalah akun Schfess, Subschfess, dan Sbmqtnfess. Sedangkan, *key actors* pada jaringan Zenius adalah akun Schfess dan Zenius_oliv.

5.2. Saran

Berdasarkan hasil analisis yang telah dilakukan, terdapat beberapa saran bagi perusahaan:

1. Akun Twitter perusahaan dapat lebih aktif lagi melakukan *engagement* atau interaksi dengan *followers* yang dimiliki. Sehingga percakapan antara pengguna Twitter mengenai perusahaan tersebut, tidak hanya mengalami peningkatan tren, tetapi juga dapat membentuk jaringan sosial dengan struktur yang unggul dibandingkan perusahaan pesaing.
2. Perusahaan juga dapat melakukan kerja sama dengan *key actors*, sehingga dapat menyebarkan informasi lebih cepat dan luas di Twitter. Setelah ditelusuri lebih lanjut, akun-akun pengguna yang merupakan *key actors* pada jaringan Ruangguru dan Zenius adalah Schfess yang merupakan akun-akun komunitas anak sekolah atau pelajar di Indonesia yang berisi informasi seputar dunia pendidikan.
3. Perusahaan juga dapat menyertakan *stakeholders* dalam berinteraksi dengan pengguna Twitter. Salah satu pengguna yang mempunyai nilai metrik *centrality* yang cukup tinggi pada masing-masing jaringan adalah pengguna dengan nama akun sabdaps dan imanusman. Setelah ditelusuri lebih lanjut, kedua akun tersebut merupakan *founder* dari Zenius dan Ruangguru. Hal ini menunjukkan bahwa cukup banyak pengguna Twitter yang berinteraksi dengan sosok yang merepresentasikan perusahaan.

Saran yang dapat diberikan untuk perbaikan pada penelitian selanjutnya antara lain:

1. Mengambil data dengan kata kunci yang lebih variatif dan memperpanjang rentang waktu pengambilan data.
2. Mengambil atau menambah data dari media sosial atau *website* lainnya.
3. Melakukan analisis jaringan sosial dengan data yang *real-time*.
4. Dapat menerapkan metode SNA pada studi kasus yang berbeda dengan menambahkan metrik dari metode-metode lainnya.
5. Dapat menganalisis jaringan sosial dengan berbagai metode lainnya dalam *social media analytics*.

DAFTAR PUSTAKA

- Alamsyah, A., Putri, F., & Omar Sharif, O. (2014). *Social Network Modelling Approach for Brand Awareness*. <https://doi.org/10.13140/2.1.4533.3441>
- Antoniadis, I., & Charmantzi, A. (2016). Social network analysis and social capital in marketing: theory and practical implementation. *International Journal of Technology Marketing*, 11, 344. <https://doi.org/10.1504/IJTMKT.2016.077387>
- DataReportal. (2022). *Digital in Indonesia: All the Statistics You Need in 2021 — Global Digital Insights*. <https://datareportal.com/reports/digital-2021-indonesia>
- Fornito, A., Zalesky, A., & Bullmore, E. (2016). *Fundamentals of brain network analysis*. Academic Press.
- Grandjean, M. (2021). *Introduction to Social Network Analysis: Basics and Historical Specificities*. <https://doi.org/10.5281/ZENODO.5083036>
- Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.
- Hartsfield, N., & Ringel, G. (2013). *Pearls in graph theory: a comprehensive introduction*. Courier Corporation.
- Himelboim, I., & Golan, G. (2019). A Social Networks Approach to Viral Advertising: The Role of Primary, Contextual, and Low Influencers. *Social Media + Society*, 5, 205630511984751. <https://doi.org/10.1177/2056305119847516>
- Litterio, A., Nantes, E. A., Larrosa, J., & Gómez, L. (2017). Marketing and social networks: a criterion for detecting opinion leaders. *European Journal of Management and Business Economics*, 26, 347–366. <https://doi.org/10.1108/EJMBE-10-2017-020>
- Mollett, A., Moran, D., & Dunleavy, P. (2011). *Using Twitter in university research, teaching and impact activities*.
- Muff, S., Rao, F., & Caflisch, A. (2005). Local modularity measure for network clusterizations. *Physical Review. E, Statistical, Nonlinear, and Soft Matter Physics*, 72, 056107. <https://doi.org/10.1103/PhysRevE.72.056107>
- NetworkX 2.8.4 documentation (2022). <https://networkx.org/documentation>
- Network Science by Albert-László Barabási. (2016). <http://networksciencebook.com/>
- Oliveira, M., & Gama, J. (2012). An overview of social network analysis. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2. <https://doi.org/10.1002/widm.1048>
- Riaz, B., Yarrow, N., & Cali, M. (2020). EdTech in Indonesia: Ready for Take-off. *World Bank, Washington DC*.
- Ruangguru. (2020). *Ruangguru Tutup Tahun 2020 dengan Melayani Lebih dari 22 Juta Pengguna di Indonesia*. <https://www.ruangguru.com/blog/ruangguru-tutup-tahun-2020-dengan-melayani-lebih-dari-22-juta-pengguna-di-indonesia>
- Scott, J., & Stokman, F. (2015). Social Networks. *International Encyclopedia of the Social & Behavioral Sciences*. <https://doi.org/10.1016/B978-0-08-097086-8.32101-8>
- Soboleva, A., Burton, S., & Khan, A. (2015). *Marketing with Twitter: Challenges and Opportunities* (pp. 1–39). <https://doi.org/10.4018/978-1-4666-8408-9.ch001>
- Tracxn (2022). *EdTech Startups in Indonesia*. <https://tracxn.com/explore/EdTech-Startups-in-Indonesia>
- Zenius. (2020). *Merayakan 17 Tahun Zenius*. <https://www.zenius.net/blog/merayakan-17-tahun-zenius>

LAMPIRAN

Lampiran 1

Tabel Raw Data Hasil Proses Data Collection

<i>ID</i>	<i>Tweet</i>	<i>Date</i>	<i>User_ID</i>	<i>Username</i>	<i>Reply_To</i>
1410282650074554369	Akhirnya, gw memutuskan mengubah pilihan dari yang FK pil 1 UGM & 2 Unand hingga pil 1 FK Unand dan pil 2 Farmasi Unand. Belajar giatlah gw kan di GO, Eduka, Zenius, dsb. Kuliah juga tetap jalan.	2021-07-01	1266397291658674182	khairilfaiz_	[]
1410282771579162628	sch! kakak kakak angkatan 21 apakah ada yang mau hibahin akun zenius atau pahamify? aku butuh buat nyicil belajar sbmptn nih, terima kasihh	2021-07-01	1201673806156161025	schfess	[]
1410283646355853319	ini zenius aktiva gada promo gedde"an apa yakk	2021-07-01	909272182416596992	fragile33x	[]
⋮	⋮	⋮	⋮	⋮	⋮
1443361925060972544	ada yang berminat sharing zenius sekolah+utbk+um ? sharing berdua aja ya biar lebih fokus. nanti perorang cuma ±260k, prefer soshum ya, karena aku saintek (biar pas to, gak bentrok). minat? langsung dm	2021-09-30	1226121099810729984	halloakuester	[]
1443326314425901057	@NotLofigirl @sabdaps Beli 2009. Itu tahun pertama Zenius ngeluarin produknya	2021-09-30	74468796	js_khairan	[[{'screen_name': 'NotLofigirl', 'name': 'Pinkage', 'id': '1075941328989515777'}, {'screen_name': 'sabdaps', 'name': 'Sabda PS ID', 'id': '195799559'}]]
1443321596039421953	@utbkfess English Academy by Ruangguru, yuks DM aja ada voucher beasiswaanya	2021-09-30	1412170707824365650	kodis_ruangguru	[[{'screen_name': 'utbkfess', 'name': 'CEK PINNED YUK! — UTBKFESS', 'id': '1241016981865693186'}]]

Lampiran 2

Tabel Deskripsi Atribut pada *Raw Data*

Atribut	Deskripsi
<i>Id</i>	ID dari <i>tweet</i>
<i>Conversation_id</i>	ID dari <i>tweet</i> tipe percakapan
<i>Created_at</i>	Waktu keseluruhan saat mengirim <i>tweet</i>
<i>Date</i>	Tanggal saat mengirim <i>tweet</i>
<i>Time</i>	Waktu saat mengirim <i>tweet</i>
<i>Timezone</i>	Zona waktu dari user yang mengirim <i>tweet</i>
<i>User_id</i>	ID dari <i>user</i> yang mengirim <i>tweet</i>
<i>Username</i>	<i>Username</i> dari <i>user</i> yang mengirim <i>tweet</i>
<i>Name</i>	Nama dari <i>user</i> yang mengirim <i>tweet</i>
<i>Tweet</i>	Isi dari <i>tweet</i>
<i>Language</i>	Bahasa dari <i>tweet</i>
<i>Mentions</i>	<i>User</i> yang disebutkan dalam <i>tweet</i>
<i>Urls</i>	URL yang disebutkan dalam <i>tweet</i>
<i>Photos</i>	Media foto yang diunggah dalam <i>tweet</i>
<i>Replies_count</i>	Banyaknya <i>user</i> yang membalas <i>tweet</i>
<i>Retweets_count</i>	Banyaknya <i>user</i> yang <i>retweet</i>
<i>Likes_count</i>	Banyaknya <i>user</i> yang menyukai <i>tweet</i>
<i>Hashtags</i>	Frasa kata kunci dengan tagar dalam <i>tweet</i>
<i>Link</i>	URL yang merujuk pada <i>tweet</i>
<i>Video</i>	Media video yang diunggah dalam <i>tweet</i>
<i>Reply_to</i>	User yang membalas <i>tweet</i>
<i>Geo</i>	Lokasi yang berasosiasi dengan <i>tweet</i>
<i>Translate</i>	Hasil terjemahan dari <i>tweet</i>
<i>Cashtags</i>	Frasa kata kunci dengan tanda dollar di <i>tweet</i>

Lampiran 3

Tabel *Edge List* Ruangguru

<i>Source</i>	<i>Target</i>	Jumlah Interaksi
7chillboo	kjnchsolo	1.440
morphoflies	junkykime	1.040
exosalien	junkykime	1.000
mactaadid	kyutieshii	550
yourarians	sbmptnfess	360
mumarisatulhk	schfess	360
dian_renataa	schfess	350
helloyon4	sbmptnfess	340
guidance204	schfess	340
syafiranurainun	schfess	270
guidance204	subschfess	250
mumarisatulhk	subschfess	220
jeinneblackpink	subschfess	210
dian_renataa	subschfess	210
⋮	⋮	⋮
hasahihashi	hyunsuksis_	10
hasahihashi	bxxxyedam	10
hasahihashi	ailurocfie	10
harvkvsvk	swidays7	10
harvezmoon	taytawanreal	10
haruvairy	yoshaurs	10
harutodongsaeng	misellia_	10
harutodongsaeng	justjustinpark	10
harutobaby__	reinx_	10
harutoair	ambisfs	10
zzzcapricorn	convomf	10

Lampiran 4

Tabel *Edge List* Zenius

<i>Source</i>	<i>Target</i>	Jumlah Interaksi
helloyoon4	sbmptnfess	340
helloyoon4	utbkfess	210
byunpov	subschfess	150
hopefullyperf	subschfess	140
byunpov	schfess	110
lizzypeachyy	schfess	100
zenius_oliv	subschfess	90
gabriux1	dinges_zenius	90
zeniusambis	schfess	80
hopefullyperf	schfess	80
helloyoon4	subschfess	80
zenius_oliv	schfess	70
hopefullyperf	sbmptnfess	60
keyystudies	subschfess	50
⋮	⋮	⋮
hijustcallwawa	notyourex_x_x_	10
honeyberryys	smkfess	10
homerunballcola	schfess	10
hoezxvirgo	denmanly	10
hobimakancanai	bertanyarl	10
hngrenjun__	tkmwe2wkcbytzlv	10
hjsl0ve	flowwlinn	10
hisspiyaya	subschfess	10
hisammula	schfess	10
hiresapps	prkdlx	10
hiraahero	itzjaraaa	10

Lampiran 5

Tabel Metrik *Centrality* pada Jaringan Ruangguru

No.	Pengguna	DC	BC	CC	EC
1	schfess	0.092	1997242.995	0.182	0.612
2	subschfess	0.064	1349604.866	0.170	0.286
3	ambisfs	0.060	843058.766	0.146	0.138
4	sbmptnfess	0.045	884745.103	0.165	0.121
5	guidance204	0.019	395194.528	0.165	0.072
6	dian_renataa	0.015	282605.591	0.163	0.067
7	paniijjekhyuk	0.012	418155.367	0.106	0.000
8	bertanyarl	0.010	173097.511	0.132	0.007
9	convomf	0.010	176510.194	0.146	0.021
10	rut0w0rld_	0.009	199741.383	0.093	0.000
11	utbkfess	0.009	241895.408	0.153	0.037
12	sabdaps	0.009	153180.923	0.118	0.001
13	zenius_oliv	0.009	230035.829	0.156	0.053
14	hopefullyperf	0.008	183037.786	0.154	0.050
15	schfess	0.008	170867.502	0.161	0.058
⋮	⋮	⋮	⋮	⋮	⋮
5.479	beerdebuuu	0.000	0.000	0.000	0.000
5.480	yayoungiee	0.000	0.000	0.000	0.000
5.481	tresno_arto	0.000	0.000	0.000	0.000
5.482	ydelfay	0.000	0.000	0.000	0.000
5.483	highekspektasi	0.000	0.000	0.000	0.000
5.484	wtfjaktim	0.000	0.000	0.000	0.000
5.485	thaiteastudy	0.000	0.000	0.000	0.000
5.486	zlaraaa27	0.000	0.000	0.000	0.000
5.487	asdfghjkioveyou	0.000	0.000	0.000	0.000
5.488	yparkdam	0.000	0.000	0.000	0.000

Lampiran 6

Tabel Metrik *Centrality* pada Jaringan Zenius

No.	Pengguna	DC	BC	CC	EC
1	schfess	0.141	483501.500	0.206	0.694
2	sbmptnfess	0.079	260140.353	0.177	0.066
3	subschfess	0.071	248605.167	0.174	0.109
4	sabdaps	0.019	60762.333	0.132	0.002
5	zenius_oliv	0.018	134208.989	0.197	0.054
6	bertanyarl	0.017	55184.864	0.148	0.005
7	hopefullyperf	0.017	101940.437	0.190	0.051
8	convomf	0.014	42055.356	0.141	0.010
9	utbkfess	0.013	64103.409	0.147	0.011
10	byunpov	0.009	57172.050	0.185	0.049
11	zenambis	0.007	26876.694	0.148	0.038
12	ambisfs	0.007	20555.743	0.134	0.009
13	vickyutami2	0.007	14675.470	0.140	0.007
14	gapyearfess	0.006	14597.241	0.144	0.006
15	keyystudies	0.004	16830.344	0.162	0.043
⋮	⋮	⋮	⋮	⋮	⋮
2.595	sumberprotein	0.000	0.000	0.000	0.000
2.596	cryptosanthoshg	0.000	0.000	0.000	0.000
2.597	tupfai	0.000	0.000	0.000	0.000
2.598	kokoradenmogu	0.000	0.000	0.000	0.000
2.599	ytaeluvv	0.000	0.000	0.000	0.000
2.600	physiciansoon	0.000	0.000	0.000	0.000
2.601	ytanakamo	0.000	0.000	0.000	0.000
2.602	pramidew	0.000	0.000	0.000	0.000
2.603	mahirahaul	0.000	0.000	0.000	0.000
2.604	sweety0ngg	0.000	0.000	0.000	0.000

Lampiran 7

Tabel *Modularity Class* pada Jaringan Ruangguru

Pengguna	<i>Modularity Class</i>
7chillboo	0
kjnchsolo	0
morphoflies	1
junkyukime	1
exosalien	1
mactaadid	2
kyutieshii	2
yourarians	3
sbmptnfess	4
mumarisatulhk	5
schfess	6
dian_renataa	3
helloyoon4	81
guidance204	8
:	:
hayitanis	4
haurucastle	9
hataraca	31
gistudees	31
ailurocfie	650
harvezmoon	32
taytawanreal	32
haruvairy	98
harutobaby__	888
harutoair	10
zzzcapricorn	9

Lampiran 8

Tabel *Modularity Class* pada Jaringan Zenius

Pengguna	<i>Modularity Class</i>
helloyon4	0
sbmptnfess	20
utbkfess	0
byunpov	29
subschfess	3
hopefullyperf	95
schfess	5
lizzypeachyy	20
zenius_oliv	0
gabriux1	7
dinges_zenius	7
zeniusambis	0
keyystudies	9
maillov_	5
:	:
flowwlinn	223
hisspiyaya	3
hisammula	5
hiresapps	239
hiraahero	20
hipokritx	5
hinumaaa	32
hilmiluthfi_	16
flammable__	16
zxcvopw	17
akutehbulan	17

BIODATA PENULIS



Adrian Maulana Muhammad, atau biasa dipanggil Adrian, lahir di Ujung Pandang pada tanggal 18 Juli 1996. Pendidikan formal yang sudah ditempuh oleh penulis dimulai dari TK Antam Pomalaa, SDI Al-Ikhlas Jakarta, SMPN 85 Jakarta, dan SMAN 34 Jakarta. Untuk saat ini penulis sedang menempuh Pendidikan S1 di Departemen Matematika, Fakultas Sains dan Analitika Data, Institut Teknologi Sepuluh Nopember. Selama berkuliah penulis mengambil bidang minat Ilmu Komputer, khususnya bidang Pembelajaran Mesin dan Big Data. Selama kuliah, penulis aktif mengikuti kepanitiaan event, diantaranya OMITS (Olimpiade Matematika ITS) sebagai Ketua Koordinator Akomodasi dan Transportasi (2017). Penulis juga mengikuti beberapa pelatihan pada bidang ilmu komputer, salah satunya adalah Google Developers Kejar (2019). Selain itu penulis juga aktif dalam berbagai pelatihan kemahasiswaan seperti LKMM Pra-TD, LKMM TD, pelatihan PKM GT, Mathematics Heroes School, dsb. Dalam penulisan Tugas Akhir ini tidak lepas dari kekurangan, untuk itu penulis mengharapkan kritik dan saran mengenai penulisan Tugas Akhir ini yang dapat dikirimkan melalui e-mail ke adrian.m.muhammad@gmail.com.Terimakasih.