

TUGAS AKHIR - KM184801

**ANALISIS INTERAKSI PENGGUNA TWITTER
MENGENAI BISNIS *EDUCATIONAL TECHNOLOGY*
MENGUNAKAN PENDEKATAN *SOCIAL NETWORK
ANALYSIS* (STUDI KASUS: RUANGGURU DAN
ZENIUS)**

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

“Halaman ini sengaja dikosongkan”



TUGAS AKHIR - KM184801

**ANALISIS INTERAKSI PENGGUNA TWITTER
MENGENAI BISNIS *EDUCATIONAL TECHNOLOGY*
MENGUNAKAN PENDEKATAN *SOCIAL NETWORK
ANALYSIS* (STUDI KASUS: RUANGGURU DAN
ZENIUS)**

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

“Halaman ini sengaja dikosongkan”



FINAL PROJECT - KM184801

**TWITTER USER INTERACTION ANALYSIS ABOUT
EDUCATIONAL TECHNOLOGY BUSINESS USING
SOCIAL NETWORK ANALYSIS APPROACH (CASE
STUDIES: RUANGGURU AND ZENIUS)**

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

“Halaman ini sengaja dikosongkan”

LEMBAR PENGESAHAN

ANALISIS INTERAKSI PENGGUNA TWITTER MENGENAI BISNIS *EDUCATIONAL TECHNOLOGY* MENGGUNAKAN PENDEKATAN *SOCIAL NETWORK ANALYSIS* (STUDI KASUS: RUANGGURU DAN ZENIUS)

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

“Halaman ini sengaja dikosongkan”

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 Interaksi Pengguna Twitter Mengenai Bisnis *Educational Technology* Menggunakan Pendekatan *Social Network Analysis* (Studi Kasus: Ruangguru dan Zenius” 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, 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

“Halaman ini sengaja dikosongkan”

ABSTRAK

ANALISIS INTERAKSI PENGGUNA TWITTER MENGENAI BISNIS *EDUCATIONAL TECHNOLOGY* MENGGUNAKAN PENDEKATAN *SOCIAL NETWORK ANALYSIS* (STUDI KASUS: RUANGGURU DAN ZENIUS)

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. Meningkatnya jumlah pengguna media sosial di Indonesia dapat membuka peluang baru bagi bisnis *educational technology* (*edtech*) untuk meningkatkan efektivitas kegiatan pemasarannya di media sosial. Penelitian ini mengusulkan implementasi *Social Network Analysis* (SNA) dalam menganalisis interaksi pengguna media sosial mengenai bisnis *edtech*. SNA merupakan suatu pendekatan analitis yang memanfaatkan teori graf untuk mengidentifikasi struktur suatu jaringan sosial. Metrik SNA yang digunakan pada penelitian ini adalah metrik *network properties* untuk mendeskripsikan karakteristik dari jaringan dan metrik *centrality* untuk mengidentifikasi *key actor* di dalam jaringan. Jaringan yang terbentuk berdasarkan percakapan pengguna Twitter mengenai dua perusahaan *edtech* di Indonesia, yaitu Ruangguru dan Zenius.

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

“Halaman ini sengaja dikosongkan”

ABSTRACT

TWITTER USER INTERACTION ANALYSIS ABOUT EDUCATIONAL TECHNOLOGY BUSINESS USING SOCIAL NETWORK ANALYSIS APPROACH (STUDY CASES: RUANGGURU AND ZENIUS)

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. The increasing number of social media users in Indonesia could open up new opportunities for educational technology (edtech) businesses to increase the effectiveness of their marketing activities on social media. This study proposes the implementation of Social Network Analysis (SNA) in analyzing the interaction of social media users regarding the edtech business. SNA is an analytical approach that utilizes graph theory to identify the structure of a social network. SNA metrics used in this study are network properties metrics to describe the characteristics of the network and centrality metrics to identify key actors in the network. The network was formed based on conversations by Twitter users about two edtech companies in Indonesia: Ruangguru and Zenius.

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

“Halaman ini sengaja dikosongkan”

KATA PENGANTAR

Alhamdulillahirabbil'alamin, puji syukur saya panjatkan kepada Allah SWT yang telah memberikan limpahan rahmat, petunjuk, serta hidayah-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “Analisis Interaksi Pengguna Twitter Mengenai Bisnis *Educational Technology* Menggunakan Pendekatan *Social Network Analysis* (Studi Kasus: Ruangguru dan Zenius)” 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

“Halaman ini sengaja dikosongkan”

DAFTAR ISI

LEMBAR PENGESAHAN	i
PERNYATAAN ORISINALITAS	iii
ABSTRAK	v
ABSTRACT	vii
KATA PENGANTAR	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiv
DAFTAR KODE PROGRAM	xvi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
BAB II TINJAUAN PUSTAKA	4
2.1 Hasil Penelitian Terdahulu	5
2.2 Dasar Teori	6
2.2.1 Ruangguru	6
2.2.2 Zenius	6
2.2.3 Twitter	6
2.2.4 <i>Data Mining</i>	6
2.2.5 Teori Graf	7
2.2.6 <i>Social Network Analysis (SNA)</i>	8
BAB III METODOLOGI	15
3.1. Bahan dan peralatan yang digunakan	15
3.2. Urutan pelaksanaan penelitian	15
BAB IV HASIL DAN PEMBAHASAN	18
4.1 <i>Data Collection</i>	19
4.2 <i>Data Pre-processing</i>	19
4.2.1 <i>Import Dataset</i>	20
4.2.2 Menghapus <i>Tweet</i> yang Duplikat	20
4.2.3 Menghapus <i>Tweet</i> yang Tidak Mempunyai Interaksi	20
4.2.4 Mengambil <i>Tweet</i> Percakapan Antara Pengguna	21
4.2.5 Pengelompokan <i>Tweet</i> Mengenai Perusahaan <i>Edtech</i>	21
4.2.6 Transformasi Data ke Bentuk <i>Edge List</i>	22
4.3 <i>Social Network Analysis (SNA)</i>	23
4.3.1 <i>Network Properties</i>	23
4.3.2 <i>Centrality</i>	27
4.4 Visualisasi Model Jaringan	33
BAB V KESIMPULAN DAN SARAN	37
5.1. Kesimpulan	37
5.2. Saran	37
DAFTAR PUSTAKA	39
LAMPIRAN	40
BIODATA PENULIS	48

“Halaman ini sengaja dikosongkan”

DAFTAR GAMBAR

Gambar 1.1 Pertumbuhan Sektor <i>Edtech</i> dan Tingkat Penetrasi Internet di Indonesia	1
Gambar 1.2 Jumlah Percakapan Pengguna Twitter Mengenai Ruangguru & Zenius.....	2
Gambar 2.1 Graf G	10
Gambar 2.2 Ilustrasi Algoritma Louvain	10
Gambar 2.3 Ilustrasi Algoritma <i>Breadth-First Search</i>	11
Gambar 2.4 Dua Pengguna yang Mempunyai Nilai <i>Degree Centrality</i> yang Sama.....	12
Gambar 3.1 Diagram Alir Penelitian	16
Gambar 3.1 Diagram Alir Penelitian (lanjutan).....	17
Gambar 4.1 Visualisasi Jaringan Ruangguru	34
Gambar 4.2 Visualisasi Jaringan Zenius.....	35

“Halaman ini sengaja dikosongkan”

DAFTAR TABEL

Tabel 2.1	Deskripsi Metrik <i>Network Properties</i>	12
Tabel 2.2	Deskripsi Metrik <i>Centrality</i>	14
Tabel 3.1	Spesifikasi Perangkat.....	15
Tabel 3.2	Aplikasi dan Teknologi.....	14
Tabel 4.1	Hasil Perhitungan Metrik <i>Network Properties</i> pada Jaringan Ruangguru dan Zenius	27
Tabel 4.2	Lima Pengguna dengan Nilai <i>Degree Centrality</i> Tertinggi pada Jaringan Ruangguru	28
Tabel 4.3	Lima Pengguna dengan Nilai <i>Degree Centrality</i> Tertinggi pada Jaringan Zenius	28
Tabel 4.4	Lima Pengguna dengan Nilai <i>Betweenness Centrality</i> Tertinggi pada Jaringan Ruangguru.....	29
Tabel 4.5	Lima Pengguna dengan Nilai <i>Betweenness Centrality</i> Tertinggi pada Jaringan Zenius	30
Tabel 4.6	Lima Pengguna dengan Nilai <i>Closeness Centrality</i> Tertinggi pada Jaringan Ruangguru	31
Tabel 4.7	Lima Pengguna dengan Nilai <i>Closeness Centrality</i> Tertinggi pada Jaringan Zenius	31
Tabel 4.8	Lima Pengguna dengan Nilai <i>Eigenvector Centrality</i> Tertinggi pada Jaringan Ruangguru.....	32
Tabel 4.9	Lima Pengguna dengan Nilai <i>Eigenvector Centrality</i> Tertinggi pada Jaringan Zenius	32
Tabel 4.10	Seluruh Hasil Perhitungan Metrik <i>Centrality</i> pada Jaringan Ruangguru	32
Tabel 4.11	Seluruh Hasil Perhitungan Metrik <i>Centrality</i> pada Jaringan Zenius	33
Tabel 4.12	Lima Kelompok Terbesar pada Jaringan Ruangguru	34
Tabel 4.13	Lima Kelompok Terbesar pada Jaringan Zenius	35

“Halaman ini sengaja dikosongkan”

DAFTAR KODE PROGRAM

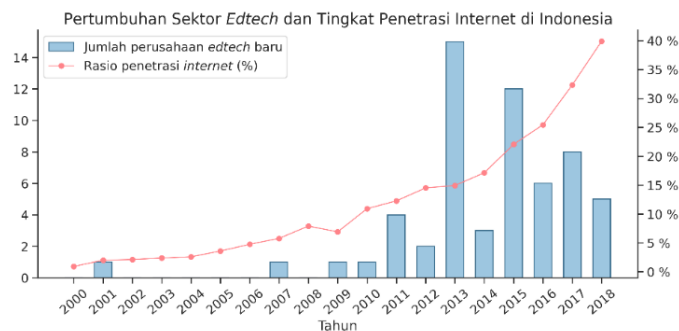
Kode Program 4.1 <i>Scraping Data Twitter</i>	19
Kode Program 4.2 <i>Import Dataset</i>	20
Kode Program 4.3 <i>Menghapus Tweet yang Duplikat</i>	20
Kode Program 4.4 <i>Menghapus Tweet yang Tidak Mempunyai Interaksi</i>	20
Kode Program 4.5 <i>Mengambil Tweet Percakapan Antara Pengguna</i>	21
Kode Program 4.6 <i>Pengelompokan Tweet Mengenai Kedua Perusahaan Edtech</i>	21
Kode Program 4.7 <i>Transformasi Data ke Bentuk Edge List</i>	22
Kode Program 4.8 <i>Transformasi Edge List ke Undirected Graph</i>	23
Kode Program 4.9 <i>Order dan Size pada Jaringan</i>	23
Kode Program 4.10 <i>Density pada Jaringan</i>	24
Kode Program 4.11 <i>Modularity pada Jaringan</i>	24
Kode Program 4.12 <i>Diameter pada Jaringan</i>	25
Kode Program 4.13 <i>Average Path Length pada Jaringan</i>	25
Kode Program 4.14 <i>Average Degree pada Jaringan</i>	26
Kode Program 4.15 <i>Connected Components pada Jaringan</i>	27
Kode Program 4.16 <i>Degree Centrality pada Jaringan</i>	28
Kode Program 4.17 <i>Betweenness Centrality pada Jaringan</i>	29
Kode Program 4.18 <i>Closeness Centrality pada Jaringan</i>	30
Kode Program 4.19 <i>Eigenvector Centrality pada Jaringan</i>	31
Kode Program 4.20 <i>Transformasi Data Graf Networkx ke File GEXF</i>	31

“Halaman ini sengaja dikosongkan”

BAB I PENDAHULUAN

1.1 Latar Belakang

Jumlah pengguna internet di Indonesia meningkat 27 juta pengguna atau sebesar 16 persen di antara tahun 2020 dan 2021. Penggunaan internet yang meningkat di Indonesia berpengaruh terhadap munculnya inovasi pada penggunaan informasi, komunikasi, dan teknologi untuk pendidikan dalam bentuk platform *educational technology* (*edtech*). Sektor bisnis *edtech* di Indonesia masih dalam fase pertumbuhan, dengan hampir seluruh perusahaan masih terus melakukan eksperimen pada produk ataupun pasar mereka. Pada Gambar 1.1 menunjukkan bahwa peningkatan pada pendirian perusahaan *edtech* juga bersesuaian dengan peningkatan penetrasi internet di Indonesia (Bhardwaj, Yarrow, & Cali, 2020).



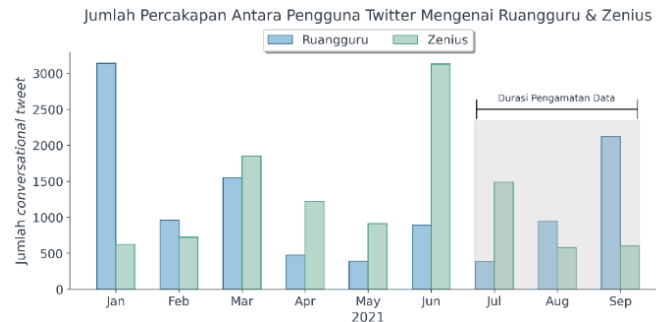
Gambar 1.1 Pertumbuhan Sektor *Edtech* dan Tingkat Penetrasi Internet di Indonesia
Sumber Data: (Bhardwaj, Yarrow, & Cali, 2020)

Dua perusahaan *edtech*, yaitu Ruangguru dan Zenius, termasuk dalam daftar perusahaan *edtech* yang menonjol dalam hal pertumbuhan pengguna dan perhatian investor selama beberapa tahun terakhir di Indonesia (Bhardwaj, Yarrow, & Cali, 2020). Selain hal tersebut, pada September 2021, Ruangguru dan Zenius juga termasuk akun platform *edtech* di Indonesia yang mempunyai *followers* terbanyak pada situs jejaring sosial *online*, Twitter. Namun, dengan memiliki jumlah *followers* Twitter yang besar, tidak berarti banyak jika tingkat interaksi rendah dan perusahaan tidak melibatkan *followers* tersebut dalam kegiatan pemasaran. Oleh karena itu, diperlukan suatu analisis yang dapat membantu perusahaan dalam memahami pola interaksi konsumen mereka di Twitter, sehingga perusahaan dapat mengidentifikasi penyebab aktivitas pemasaran produk mereka tidak seberhasil dengan perusahaan pesaing, ataupun sebaliknya.

Salah satu metode dalam *social media analytics* yang umum digunakan untuk menganalisis pola interaksi antara individu adalah *Social Network Analysis* (SNA). SNA merupakan pendekatan analitis yang memanfaatkan teori graf untuk mengidentifikasi struktur suatu jaringan sosial. Suatu jaringan sosial dapat terdiri dari pengguna Twitter, dilambangkan dengan simpul (*node*), dan interaksi antara pengguna Twitter tersebut, dilambangkan dengan sisi (*edge*).

Pada Gambar 1.2 terlihat bahwa, sekitar awal Juli sampai dengan akhir September 2021, jumlah percakapan pengguna Twitter mengenai Zenius mengalami penurunan, sedangkan percakapan mengenai Ruangguru terus meningkat, bahkan melebihi jumlah percakapan mengenai Zenius. Menganalisis pada durasi ini dapat menjawab pertanyaan apakah pergerakan *trend* jumlah percakapan pengguna Twitter mengenai kedua perusahaan tersebut berbanding lurus dengan kualitas karakteristik jaringan sosial yang terbentuk. Pada SNA, kualitas karakteristik 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 terhadap percakapan di dalam jaringan tersebut. Pada SNA, *key actor* di dalam suatu jaringan dapat diidentifikasi dengan metrik *centrality*. Durasi pengamatan data ditunjukkan pada Gambar 1.2.



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

Terdapat beberapa penelitian terdahulu yang membahas implementasi 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 ditulis oleh Arnaldo Litterio dkk. mengenai penerapan SNA dalam pemasaran untuk mengidentifikasi *opinion leaders* (Litterio, Nantes, Larrosa, & Gómez, 2017). Ketiga, penelitian yang ditulis oleh Itai Himelboim dan Guy Golan mengenai pendekatan *social networks* dalam menganalisis peran *influencer* pada *viral advertising* (Himelboim & Golan, 2019).

Data yang digunakan pada penelitian ini adalah *tweets* yang diperoleh dari Twitter dengan kata kunci: “ruangguru” dan “zenius” sejak 1 Juli 2021 sampai dengan 30 September 2021. Data yang terkumpul adalah sejumlah 39.219 *tweets*, dengan 5.488 simpul dan 4.982 sisi untuk Ruangguru dan 2.605 simpul dan 2.123 sisi untuk Zenius.

Berdasarkan latar belakang yang telah diuraikan di atas, maka penulis melakukan suatu penelitian tentang analisis interaksi pengguna Twitter mengenai dua perusahaan *edtech*, yaitu Ruangguru dan Zenius, dengan menggunakan pendekatan SNA. Metrik yang digunakan pada pendekatan SNA adalah metrik *network properties* dan *centrality*.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan di atas, permasalahan yang akan dibahas dalam penelitian ini adalah sebagai berikut:

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

1.3 Batasan Masalah

Penulis membatasi permasalahan pada penelitian ini sebagai berikut:

1. Data yang dianalisis adalah *tweets* yang diperoleh dari Twitter dengan kata kunci “ruangguru” dan “zenius” sejak 1 Juli 2021 sampai dengan 30 September 2021.

2. Jenis interaksi antara pengguna Twitter yang dianalisis adalah *reply* yang merupakan tanggapan atas suatu *tweet*.
3. Jenis graf yang digunakan adalah graf tak berarah (*undirected graph*).
4. Proses pengumpulan data dilakukan dengan metode *scraping* pada *website* Twitter yang dapat diakses bebas oleh publik.

1.4 Tujuan Penelitian

Tujuan dalam penelitian ini adalah sebagai berikut:

1. Menganalisis karakteristik 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 *centralit*.

1.5 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini sebagai berikut:

1. Penelitian ini dapat bermanfaat sebagai saran kepada kedua perusahaan di sektor *edtech*, yaitu Ruangguru dan Zenius. Sehingga, diharapkan dapat membantu kedua perusahaan tersebut dalam meningkatkan strategi pemasaran mereka di Twitter agar dapat terus berkembang dan bertahan dalam persaingan di industri *edtech*.
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

“Halaman ini sengaja dikosongkan”

BAB II TINJAUAN PUSTAKA

2.1 Hasil Penelitian Terdahulu

Terdapat beberapa penelitian terdahulu yang menjadi tinjauan pustaka 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, Nantes, Larrosa, & Gómez, 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 merupakan *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).

Berdasarkan beberapa penelitian di atas, penulis akan melakukan suatu penelitian untuk menganalisis interaksi pengguna Twitter mengenai bisnis *edtech* dengan pendekatan SNA. Pendekatan SNA yang digunakan pada penelitian ini adalah metrik *network properties* dan *centrality*. Data yang dianalisis berdasarkan percakapan di Twitter mengenai dua perusahaan *edtech*, yaitu Ruangguru dan Zenius.

2.2 Dasar Teori

2.2.1 Ruangguru

Ruangguru adalah aplikasi belajar *online* yang menyediakan layanan berupa bimbingan belajar untuk membantu peserta didik dalam memahami materi pelajaran. Sampai dengan saat ini, 75 persen total pelajar di Indonesia telah menggunakan aplikasi Ruangguru untuk mengakses beragam mata pelajaran mulai dari jenjang SD, SMP, SMA, dan SMK yang dikemas secara lebih interaktif (Ruangguru, 2021).

2.2.2 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 (Zenius, 2021).

2.2.3 Twitter

Twitter adalah situs jejaring sosial *online* yang memungkinkan penggunanya mengirim dan membaca pesan berupa teks dengan panjang maksimum 280 karakter. Terdapat beberapa istilah yang digunakan dalam Twitter, yaitu *timeline* adalah daftar *tweet* terbaru dari pengguna yang diikuti oleh pemilik akun dan termasuk *tweet* yang dibuat oleh pemilik akun itu sendiri, *tweet* adalah pesan yang ditulis dan dipublikasikan oleh pengguna, *reply* adalah tanggapan atau balasan atas suatu *tweet* yang diberikan untuk pembuat *tweet* tersebut, *retweet* adalah mempublikasikan kembali suatu *tweet* dari pengguna, *mention* merupakan *tweet* yang berisi nama pengguna, *hashtag* digunakan untuk mengindeks suatu kata kunci untuk topik diskusi yang dibagikan agar mudah dicari, *follow* yaitu mengikuti akun dan informasi yang disampaikan oleh seorang pengguna, *follower* adalah pengikut atau yang mengikuti akun pengguna (Mollett, Moran, & Dunleavy, 2011).

2.2.4 Data Mining

Pada industri, media, dan lingkungan penelitian, istilah *data mining* sering digunakan untuk merujuk pada seluruh proses penemuan pengetahuan atau biasa disebut *Knowledge Discovery from Data* (KDD). Proses penemuan pengetahuan 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)

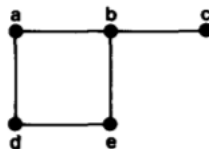
Langkah 1 hingga 4 adalah berbagai bentuk dari *data pre-processing*, di mana mempersiapkan data sebelum proses *mining*. Pola yang menarik disajikan kepada pengguna dan dapat disimpan sebagai suatu *insights* atau pengetahuan. Oleh karena itu, *data mining* adalah proses menemukan pola yang menarik dan pengetahuan dari sejumlah besar data. Sumber data dapat mencakup *database*, *data warehouses*, *web*, repositori informasi lain, atau data yang dialirkan ke sistem secara dinamis (Han, Pei, & Kamber, 2011).

2.2.5 Teori Graf

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 adalah himpunan (mungkin kosong) dari pasangan tak beraturan dari elemen V . Elemen V disebut simpul dari G dan elemen dari E disebut sisi dari G . Terkadang kita akan menulis $V(G)$ untuk simpul-simpul G dan $E(G)$ untuk sisi-sisi G . Simpul-simpul biasanya direpresentasikan sebagai titik-titik pada bidang, dan suatu sisi biasanya direpresentasikan sebagai suatu garis yang menghubungkan dua simpul pada bidang (Hartsfield & Ringel, 2013). Sehingga, Graf G dapat direpresentasikan sebagai berikut:

$$G = (V, E) \quad (2.1)$$

Berikut adalah suatu contoh ilustrasi dari graf G ditunjukkan pada Gambar 2.1:



Gambar 2.1 Graf G

(Sumber: (Hartsfield & Ringel, 2013))

Pada Gambar 2.1, dapat dilihat bahwa Graf G memiliki 5 simpul dan 5 sisi. Adapun himpunan V dan E dari Graf G adalah sebagai berikut:

$$V(G) = \{a, b, c, d, e\} \quad (2.2)$$

$$E(G) = \{(a, b), (a, d), (b, e), (b, c), (d, e)\} \quad (2.3)$$

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. Dalam suatu graf, dimungkinkan adanya lebih dari satu sisi yang menempel (*incident*) dengan sepasang simpul. Pasangan sisi semacam ini disebut sisi-sisi paralel atau sisi rangkap (*multiple edges*).
3. Suatu sisi yang hanya menempel dengan satu simpul disebut gelang (*loop*).

4. Suatu graf yang tidak memiliki gelang dan tidak memiliki sisi rangkap disebut graf sederhana (*simple graph*).
5. Derajat (*degree*) pada suatu simpul adalah banyaknya sisi yang menempel dengan simpul tersebut. Suatu simpul yang memiliki derajat 0 disebut simpul yang terisolasi, sedangkan suatu simpul yang memiliki derajat 1 disebut simpul ujung.
6. 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 .
7. 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*).
8. 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.

2.2.6 Social Network Analysis (SNA)

Social Network Analysis (SNA) merupakan suatu pendekatan analitis yang dapat mengidentifikasi struktur suatu jaringan sosial dengan memanfaatkan teori graf. Jaringan sosial yang terbentuk dapat terdiri dari pengguna, dilambangkan dengan simpul, dan interaksi antara pengguna, dilambangkan dengan sisi. SNA diperlukan karena membawa kesempatan baru untuk memahami individu atau kelompok terkait pola interaksi mereka. Pada implementasi sebelumnya, penelitian dengan pendekatan SNA menggunakan metode wawancara dan observasi untuk mendapatkan informasi tentang kualitas interaksi sosial pada suatu komunitas. Untuk menghindari kerumitan, penelitian SNA terdahulu biasanya dilakukan pada komunitas terbatas (Network Science, 2021).

Perkembangan teknologi komputasi saat ini telah memungkinkan peneliti untuk memproses kumpulan data dengan ukuran yang besar, ribuan atau bahkan ratusan ribu simpul dan sisi dapat diproses dengan menggunakan teknologi komputasi. Namun, ketika ukuran jaringan terlalu besar, menganalisis dengan pendekatan visualisasi saja sering dianggap tidak efektif. Oleh karena itu, dibutuhkan metrik-metrik yang dapat mendeskripsikan karakteristik jaringan sosial yang terbentuk. SNA menggunakan pendekatan dari teori graf untuk mendeskripsikan karakteristik jaringan yang nantinya dapat digunakan sebagai metrik. Pada penelitian ini, penulis membagi metrik menjadi dua jenis, yaitu *network properties* dan *centrality* (Network Science, 2021).

2.2.6.1 Network Properties

SNA memiliki beberapa metrik *network properties* yang berguna untuk mendeskripsikan karakteristik 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*.

2.2.6.1.1 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 menunjukkan banyaknya pengguna yang berinteraksi di jaringan tersebut. Dalam pembahasan berikutnya, banyaknya simpul akan

direpresentasikan sebagai variabel n dan banyaknya sisi direpresentasikan sebagai variabel m (Network Science, 2021).

2.2.6.1.2 Density

Density merupakan perhitungan banyaknya sisi yang ada dibandingkan dengan banyaknya sisi maksimum yang mungkin ada di dalam suatu jaringan. *Density* menggambarkan kerapatan pada jaringan, semakin tinggi nilai *density*, maka semakin baik karena menggambarkan bahwa pengguna-pengguna yang ada di dalam jaringan lebih saling terhubung antara satu sama lain. Formula untuk menghitung *density* adalah sebagai berikut (Network Science, 2021):

$$\text{Network Density} = \frac{\text{Total Edges}}{\text{Total Possible Edges}} = \frac{m}{n(n-1)/2} \quad (2.4)$$

keterangan:

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

2.2.6.1.3 Modularity

Modularity merupakan metrik yang digunakan untuk mengetahui kualitas pembagian jaringan ke dalam kelompok atau kluster. Formula untuk mendapatkan nilai *modularity* adalah sebagai berikut (Network Science, 2021):

$$M = \frac{1}{2m} \sum_{i,j} \left[A_{i,j} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j) \quad (2.5)$$

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
- $\delta(c_i, c_j)$: Kronecker Delta

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* (Network Science, 2021).

1. Modularity Optimization

Pertama, 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 modularitas, jika tidak terdapat peningkatan nilai modularitas, 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 modularitas. Perubahan nilai modularitas (ΔM) dapat dihitung dengan formula sebagai berikut (Network Science, 2021):

$$\Delta M = \left[\frac{\sum_{in} + 2k_{i,in}}{2m} - \left(\frac{\sum_{tot} + k_i}{2m} \right)^2 \right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m} \right)^2 - \left(\frac{k_i}{2m} \right)^2 \right] \quad (2.6)$$

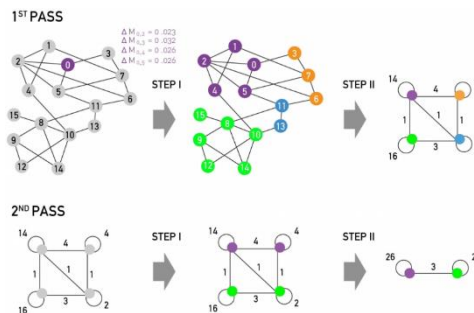
keterangan:

- \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 (Network Science, 2021).

Setelah tahapan kedua selesai, tahapan pertama sampai kedua akan diulangi lagi, dengan menyebut iterasinya sebagai *pass*. Jumlah kelompok akan berkurang pada setiap *pass*. *Pass* diulang sampai tidak ada lagi perubahan dan modularitas maksimum telah tercapai. Ilustrasi algoritma Louvain ditunjukkan pada Gambar 2.2 (Network Science, 2021).



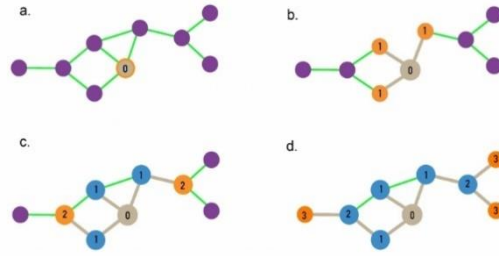
Gambar 2.2 Ilustrasi Algoritma Louvain
(Sumber: (Network Science, 2021))

2.2.6.1.4 Diameter

Diameter adalah jarak lintasan terpendek (*shortest-path*) terpanjang atau terjauh antara sepasang simpul di dalam jaringan. Semakin kecil nilai *diameter* maka semakin baik, karena proses penyebaran informasi antara suatu pengguna dengan pengguna lainnya, dengan jarak yang terjauh, hanya perlu melewati sedikit pengguna. Nilai *diameter* yang kecil mengindikasikan proses penyebaran informasi akan memakan waktu lebih sedikit karena melibatkan lebih sedikit pengguna. Untuk jaringan yang berukuran besar, lintasan terpendek dapat ditentukan menggunakan algoritma *Breadth-First Search* (BFS). *Diameter* dari suatu jaringan biasa dinotasikan sebagai d_{max} . Identifikasi lintasan terpendek antara simpul i dan simpul j dengan algoritma BFS mengikuti langkah-langkah berikut (Network Science, 2021):

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 $d_{ij} = \infty$.



Gambar 2.3 Ilustrasi Algoritma *Breadth-First Search*
(Sumber: (Network Science, 2021))

2.2.6.1.5 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. *Average path length* biasa dinotasikan sebagai $\langle d \rangle$. Kita dapat menggunakan algoritma BFS, seperti yang sudah dibahas sebelumnya, untuk menentukan lintasan terpendek di dalam jaringan. Formula untuk menghitung *average path length* adalah sebagai berikut (Network Science, 2021).

$$\langle d \rangle = \frac{1}{n(n-1)} \sum_{\substack{i,j=1,n \\ i \neq j}} d_{i,j} \quad (2.7)$$

keterangan:

- n : jumlah simpul pada jaringan
 $d_{i,j}$: jarak lintasan terpendek antara simpul i dan j

2.2.6.1.6 Average Degree

Average degree merupakan perhitungan rata-rata derajat atau rata-rata banyaknya sisi yang menghubungkan suatu simpul ke simpul lain pada suatu jaringan. Semakin besar nilai *average degree* yang dimiliki oleh jaringan maka semakin baik, karena apabila suatu pengguna menyebarkan informasi ke lebih banyak pengguna lainnya, maka akan mempercepat penyebaran informasi di dalam jaringan. Formula untuk menghitung *average degree* adalah sebagai berikut (Network Science, 2021):

$$\langle k \rangle = \frac{1}{n} \sum_{i=1}^n k_i \quad (2.8)$$

keterangan:

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

2.2.6.1.7 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 pengguna tidak banyak terpisah ke dalam kelompok-kelompok kecil yang tidak saling terkoneksi. *Connected components* dapat dicari dengan menggunakan algoritma BFS yang telah dibahas sebelumnya, karena urutan dari simpul tidak menjadi masalah (Network Science, 2021).

Deskripsi keseluruhan metrik *network properties* yang digunakan pada penelitian ini ditunjukkan pada Tabel 2.1 (Network Science, 2021).

Tabel 2.1 Deskripsi Metrik *Network Properties*

No.	<i>Network Properties</i>	Deskripsi
1.	<i>Order</i>	Jumlah simpul di jaringan.
2.	<i>Size</i>	Jumlah sisi di jaringan.
3.	<i>Density</i>	Kerapatan pada jaringan.
4.	<i>Modularity</i>	Kualitas pembagian jaringan ke dalam kelompok.
5.	<i>Diameter</i>	Jarak <i>shortest-path</i> terpanjang atau terjauh di antara sepasang simpul di jaringan.
6.	<i>Average Path Length</i>	Rata-rata jarak <i>shortest-path</i> di antara setiap pasang simpul di jaringan.
7.	<i>Average Degree</i>	Rata-rata derajat pada setiap simpul di jaringan.
8.	<i>Connected Components</i>	Kelompok simpul yang terpisah atau tidak terkoneksi dengan keseluruhan jaringan.

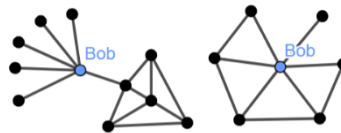
(Sumber: (Network Science, 2021))

2.2.6.2 Centrality

Setelah metrik *network properties*, metrik selanjutnya adalah *centrality*. Pengukuran pada metrik *centrality* bertujuan untuk mengidentifikasi pengguna yang paling berpengaruh (*key actor*) di antara semua pengguna yang ada di dalam suatu jaringan. Terdapat empat pengukuran *centrality* pada penelitian ini, yaitu: *degree centrality*, *betweenness centrality*, *closeness centrality*, dan *eigenvector centrality* (Fornito, Zalesky, & Bullmore, 2016).

2.2.6.2.1 Degree Centrality

Metrik *degree centrality* menggambarkan ukuran *social connections* yang dimiliki pengguna di dalam jaringan. Suatu simpul 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 2.4 menunjukkan pengguna bernama Bob yang memiliki nilai *degree centrality* yang paling tinggi 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. Hal ini menunjukkan, walaupun metrik *degree centrality* dapat mengidentifikasi simpul yang paling banyak memiliki *social connections*, tetapi metrik ini tidak selalu menunjukkan simpul yang berada di tengah jaringan (Fornito, Zalesky, & Bullmore, 2016).



Gambar 2.4 Dua Pengguna yang Mempunyai Nilai *Degree Centrality* yang Sama

Berikut adalah formula *degree centrality* untuk simpul i [11]:

$$C_D(i) = \sum_{i \neq j}^n a_{ij} \quad (2.9)$$

$$a_{ij} = \begin{cases} 1, & \text{apabila terdapat sisi di antara simpul } i \text{ dan } j \\ 0, & \text{yang lainnya} \end{cases} \quad (2.10)$$

keterangan:

n : jumlah simpul di jaringan.

2.2.6.2.2 Betweenness Centrality

Gagasan singkat pada metrik *betweenness centrality* adalah suatu metrik *centrality* yang tidak mementingkan seberapa banyak *social connections* yang dimiliki pengguna, tetapi lebih di mana posisi pengguna tersebut ditempatkan di dalam jaringan. Sebagai contoh, pada Gambar 2.4, pengguna bernama Bob yang berada di jaringan sebelah kiri merupakan pengguna yang memungkinkan untuk informasi dapat diteruskan dari kelompok kanan ke kelompok kiri, dan juga sebaliknya. Dengan demikian, pengguna tersebut sangat penting bagi penyebaran arus informasi di dalam jaringan. Hal ini yang berusaha ditangkap oleh perhitungan metrik *betweenness centrality*. Untuk menghitung nilai *betweenness centrality* pada suatu simpul i , kita dapat menghitung proporsi lintasan terpendek antara simpul j dan h yang melewati simpul i (Fornito, Zalesky, & Bullmore, 2016).

$$C_B(i) = \frac{1}{(n-1)(n-2)} \cdot \sum_{h \neq i, h \neq j, j \neq i}^n \frac{\rho_{hj}(i)}{\rho_{hj}} \quad (2.11)$$

keterangan:

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

2.2.6.2.3 Closeness Centrality

Gagasan singkat pada metrik *closeness centrality* adalah suatu perhitungan untuk mencari simpul yang paling dekat dengan semua simpul lainnya di dalam suatu 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. Formula untuk menghitung nilai *closeness centrality* pada *node i* adalah sebagai berikut (Fornito, Zalesky, & Bullmore, 2016):

$$C_c(i) = \frac{n-1}{\sum_{j \neq i} l_{ij}} \quad (2.12)$$

keterangan:

- l_{ij} : jarak lintasan terpendek antara simpul i dan j
- n : jumlah simpul di jaringan

2.2.6.2.4 Eigenvector Centrality

Gagasan singkat dari metrik *eigenvector centrality* adalah suatu perhitungan yang memperhatikan kuantitas dan juga kualitas koneksi yang dipunyai suatu pengguna 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 (Fornito, Zalesky, & Bullmore, 2016).

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

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

keterangan:

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

Selanjutnya, yaitu mencari *eigenvector* dengan menggunakan *eigenvalue* terbesar, sebagai berikut (Fornito, Zalesky, & Bullmore, 2016):

$$A\vec{v} = \lambda\vec{v} \quad (2.14)$$

$$(A - \lambda I)\vec{v} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \quad (2.15)$$

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 (Fornito, Zalesky, & Bullmore, 2016):

$$\vec{v} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad (2.16)$$

Eigenvector centrality dari simpul i dapat didefinisikan sebagai *input* ke- i di dalam *eigenvector* \vec{v} . Deskripsi keseluruhan metrik *centrality* yang digunakan pada penelitian ini ditunjukkan pada Tabel 2.2 (Fornito, Zalesky, & Bullmore, 2016).

Tabel 2.2 Deskripsi *Metrik Centrality*

No.	<i>Centrality</i>	Deskripsi
1.	<i>Degree Centrality</i>	Simpul dengan derajat tertinggi.
2.	<i>Betweenness Centrality</i>	Simpul yang berada di tengah simpul lainnya.
3.	<i>Closeness Centrality</i>	Simpul yang paling dekat dengan simpul lainnya.
4.	<i>Eigenvector Centrality</i>	Simpul dengan kualitas koneksi terbaik.

(Sumber: (Fornito, Zalesky, & Bullmore, 2016))

BAB III METODOLOGI

3.1. Bahan dan peralatan yang digunakan

Penelitian ini menggunakan perangkat keras dan perangkat lunak untuk menunjang proses pengerjaan. Untuk spesifikasi perangkat keras dan perangkat lunak yang digunakan ditunjukkan pada Tabel 3.1.

Tabel 3.1 Spesifikasi Perangkat

Perangkat	Lenovo IdeaPad S340
Prosesor	AMD Ryzen 3 3200U Radeon Vega Mobile Gfx 2.60 GHz
RAM	8,00 GB
Sistem Operasi	Windows 11

Pada penelitian ini juga digunakan beberapa aplikasi dan teknologi seperti perangkat lunak visualisasi graf, *code editor*, bahasa pemrograman, dan *library* yang ditunjukkan pada Tabel 3.2.

Tabel 3.2 Aplikasi dan Teknologi

Perangkat Lunak Visualisasi Graf	Gephi 0.9.2
Code Editor	Visual Studio Code 1.63
Bahasa Pemrograman	Python 3.8.10
Library	<ul style="list-style-type: none">•Pandas•Twint•Asyncio•Networkx•Community

3.2. Urutan pelaksanaan penelitian

Langkah-langkah sistematis yang akan dilakukan dalam proses pengerjaan penelitian ini adalah 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.
2. **Identifikasi Permasalahan**
Identifikasi permasalahan adalah tahapan yang dilakukan untuk merumuskan permasalahan yang akan diteliti, yaitu bagaimana penerapan SNA dalam menganalisis interaksi pengguna Twitter mengenai Ruangguru dan Zenius.
3. **Data Collection**
Pengumpulan data dilakukan dengan cara *scraping* dari media sosial Twitter. Tahapan ini 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 *tweet* yang sudah terkumpul, kemudian dilakukan tahap *data cleaning* untuk menghilangkan *tweet* yang tidak relevan agar lebih mudah untuk selanjutnya dilakukan proses analisis. Lalu, dilakukan tahap *data transformation* ke dalam bentuk *edge list*. Tahapan ini dilakukan dengan bahasa pemrograman Python dan *library* Pandas.

5. Analisis *Network Properties*

Data *edge list* kemudian diproses menggunakan *library* Networkx dan Community dari bahasa pemrograman Python untuk menghitung nilai *properties* yang dimiliki jaringan. Adapun metrik *network properties* yang akan dihitung nilainya adalah: *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 pengguna di dalam jaringan. Perhitungan metrik *centrality* yang dilakukan yaitu: *degree centrality*, *betweenness centrality*, *closeness centrality*, dan *eigenvector centrality*.

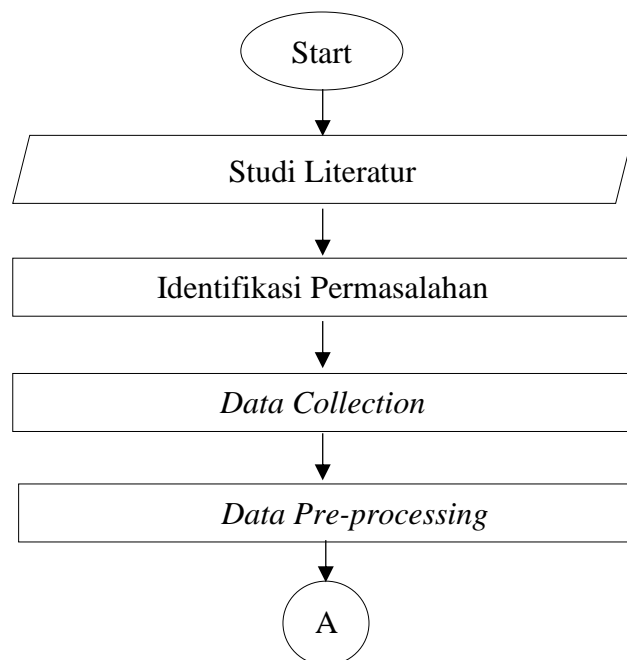
7. Visualiasi Model Jaringan

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

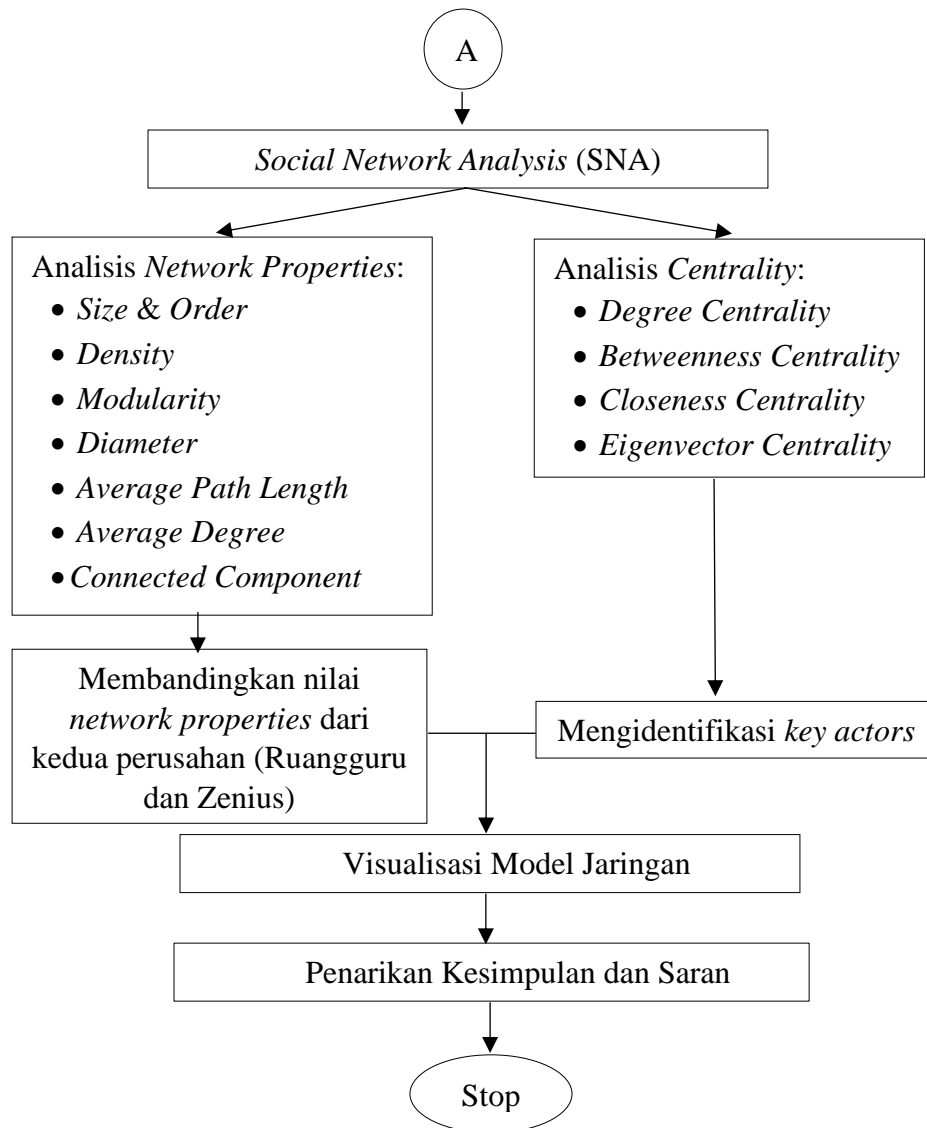
8. Kesimpulan dan Saran

Pada tahap ini, penulis melakukan penarikan kesimpulan berdasarkan hasil yang telah didapatkan dari tahapan-tahapan sebelumnya. Kemudian, memberikan saran kepada perusahaan (Ruangguru dan Zenius) dan juga kepada peneliti selanjutnya untuk mengembangkan penelitian yang telah dilakukan oleh penulis.

Langkah-langkah penelitian juga dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian



Gambar 3.2 Diagram Alir Penelitian (lanjutan)

“Halaman ini sengaja dikosongkan”

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 Comma-Separated Values* (CSV). Untuk melakukan *web scraping* maka dibutuhkan *web scraper*. *Web scraper* 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. Twint adalah suatu *web scraper* untuk mengunduh konten atau mengekstrak data *tweet* dari media sosial Twitter. Kelebihan menggunakan Twint adalah *web scraper* ini dapat mengambil hampir seluruh *tweet* (tanpa harus dibatasi seperti batasan *tweet* yang dapat diunduh hanya dalam rentang satu minggu terakhir saja), tentu saja data yang diunduh oleh Twint adalah data yang bersifat publik dari Twitter, sehingga data tersebut dapat diakses secara bebas oleh publik.

```
import twint
import nest_asyncio
nest_asyncio.apply()
tw = twint.Config()
tw.Search = "ruangguru OR zenius"
tw.Since = "2021-07-01"
tw.Until = "2021-09-30"
tw.Store_csv = True
tw.Output = "dataset.csv"
twint.run.Search(tw)
```

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 36 kolom.

4.2 Data Pre-processing

Setelah mendapatkan data mentah, maka tahapan selanjutnya yang harus dilakukan sebelum tahap analisis adalah *data pre-processing*. *Data pre-processing* merupakan teknik awal dalam *data mining* 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 *tweet* 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 *tweet* 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 (Pandas, 2022).

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 *list* tersebut tidak mempunyai elemen atau kosong, maka *tweet* tidak mempunyai interaksi. Tahapan ini dilakukan karena *tweet* yang tidak mempunyai interaksi dapat dianggap sebagai simpul yang *self-loop*. Sehingga, penghapusan *tweet* dengan kriteria ini 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 akan mengapus *tweet* yang berkaitan langsung dengan akun resmi perusahaan Ruangguru dan Zenius. Penulis hanya akan mengambil *tweet* yang merupakan percakapan antara pengguna.

```
# 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 consument 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 dari perusahaan Ruangguru dan Zenius, sehingga *dataframe* hanya berisi *tweet* percakapan antara konsumen atau pengguna biasa. Penulis menggunakan atribut *reply_to* dari *dataframe*. Pada atribut ini terdapat informasi mengenai nama dan *id* akun yang berinteraksi dengan *tweet*. Tahapan ini dilakukan karena *tweet* yang akan dianalisis adalah interaksi percakapan hanya antara pengguna.

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, sebelum menganalisis masing-masing jaringan sosial yang terbentuk, *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*

Langkah terakhir pada tahapan ini adalah melakukan transformasi pada kedua *dataframes* ke bentuk *edge lists*. *Edge list* merupakan 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 adanya interaksi percakapan atau dapat dinotasikan sebagai sisi antara suatu simpul dengan simpul lainnya. Sehingga, arah interaksi atau urutan simpul pada *edge list* 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 metrik pada SNA menjadi dua jenis, yaitu *network properties* dan *centrality* (NetworkX , 2022).

4.3.1 Network Properties

Metrik *network properties* berguna untuk mendefinisikan karakteristik 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 sedangkan nilai *density* untuk jaringan Zenius adalah 0,00062594. 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.

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 (Community detection for NetworkX, 2022).

```
import community  
# modularity function  
def modularity(g):  
    mod = community.modularity(  
        community.best_partition(  
            g, resolution=1), 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

kluster yang terbentuk dalam jaringan percakapan mengenai Zenius memiliki struktur yang secara relatif lebih baik dibandingkan pada jaringan percakapan mengenai Ruangguru.

4.3.1.4 Diameter

Untuk menghitung nilai *diameter* atau jarak *shortest-path* 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 *shortest-path* pada seluruh pasang simpul di jaringan menggunakan algoritma *Breadth-First Search* (BFS). Selanjutnya, dari seluruh jarak *shortest-path* tersebut, dicari jarak yang paling besar menggunakan fungsi *max* dari bahasa pemrograman Python. Nilai *diameter* untuk jaringan Ruangguru adalah 19 dan nilai *diameter* 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 pada jaringan percakapan mengenai Zenius memakan waktu secara relatif lebih sedikit dibandingkan pada jaringan percakapan mengenai Ruangguru karena hanya melibatkan lebih sedikit simpul.

4.3.1.5 Average Path Length

Untuk menghitung nilai *average path length* atau rata-rata *shortest-path* pada setiap pasang simpul di jaringan, penulis menggunakan fungsi *all pairs shortest-path length* dan *order* dari *library* Networkx dan fungsi *sum* dari bahasa pemrograman Python.

```
# average path length function
def avg_path(g):
    return sum(
        [sum(j.values())
         for (i,j)
         in nx.all_pairs_shortest_path_length(
             g)] / (g.order() * (g.order() - 1))
    )
# 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. Pertama, penulis mencari jarak lintasan terpendek pada seluruh pasang simpul di dalam jaringan dengan menggunakan algoritma BFS. Selanjutnya, dari seluruh jarak lintasan terpendek tersebut, dicari nilai rata-ratanya dengan menggunakan fungsi *sum* untuk mencari nilai totalnya lalu dibagi dengan banyaknya simpul pada jaringan dengan menggunakan fungsi *order*. Nilai *average path length* untuk jaringan Ruangguru adalah 1,7017 dan nilai *average path length* untuk jaringan Zenius adalah 0,94929. 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. Pertama, penulis mencari nilai derajat pada setiap simpul di jaringan. Selanjutnya, dari nilai tersebut, dicari nilai rata-ratanya dengan menggunakan fungsi *sum* untuk mencari nilai totalnya lalu dibagi dengan banyaknya simpul pada jaringan dengan menggunakan fungsi *order*. Nilai *average degree* untuk jaringan Ruangguru adalah 1,8156 dan nilai *average degree* 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 nilai *connected components* dari 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 saling terkoneksi dibandingkan pada jaringan percakapan mengenai Ruangguru.

Seluruh nilai metrik *network properties* pada jaringan percakapan pengguna Twitter mengenai Ruangguru dan Zenius ditunjukkan pada Tabel 4.1.

Tabel 4.1 Hasil Perhitungan Metrik *Network Properties* pada Jaringan Ruangguru dan Zenius

No.	<i>Network Properties</i>	Ruangguru	Zenius
1.	<i>Size</i>	4.982	2.123
2.	<i>Order</i>	5.488	2.605
3.	<i>Density</i>	0,00033089	0,00062594
4.	<i>Modularity</i>	0,87334	0,88822
5.	<i>Diameter</i>	19	13
6.	<i>Average Path Length</i>	1,7017	0,94929
7.	<i>Average Degree</i>	1,8156	1,6299
8.	<i>Connected Components</i>	1.022	587

Tabel 4.1 menunjukkan perbandingan seluruh hasil perhitungan metrik *network properties* pada jaringan Ruangguru dan Zenius. Terlihat bahwa karakteristik jaringan percakapan pengguna Twitter mengenai Ruangguru unggul dalam tiga kategori, yaitu pada metrik *size*, *order*, dan *average degree*. Sedangkan, karakteristik jaringan percakapan pengguna Twitter mengenai Zenius unggul dalam lima kategori, yaitu pada metrik *density*, *modularity*, *diameter*, *average path length*, dan *connected components*.

4.3.2 Centrality

Setelah metrik *network properties*, metrik selanjutnya adalah *centrality*. Pengukuran pada *centrality* bertujuan untuk mengidentifikasi pengguna yang paling berpengaruh (*key actor*) di antara semua pengguna 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 pengguna di masing-masing jaringan. *Output* dari kode di atas adalah berupa dua *dataframes* yang berisi kumpulan pengguna beserta nilai *degree centrality* yang dimiliki pengguna-pengguna tersebut pada masing-masing jaringan. Lima peringkat tertinggi pada *dataframe* untuk jaringan percakapan pengguna Twitter mengenai Ruangguru ditunjukkan pada Tabel 4.2.

Tabel 4.2 Lima Pengguna dengan Nilai *Degree Centrality* Tertinggi pada Jaringan Ruangguru

No.	Pengguna	<i>Degree Centrality</i>
1.	schfess	0,092218
2.	subschfess	0,063787
3.	ambisfs	0,060324
4.	sbmptnfess	0,045380
5.	guidance204	0,019136

Tabel 4.2 menunjukkan lima pengguna teratas pada metrik *degree centrality* untuk jaringan Ruangguru. Kelima pengguna tersebut adalah akun schfess, subschfess, ambisfs, sbmptnfess, dan guidance204. Selanjutnya, lima peringkat tertinggi pada *dataframe* untuk jaringan percakapan pengguna Twitter mengenai Zenius ditunjukkan pada Tabel 4.3.

Tabel 4.3 Lima Pengguna dengan Nilai *Degree Centrality* Tertinggi pada Jaringan Zenius

No.	Pengguna	<i>Degree Centrality</i>
1.	schfess	0,14132
2.	sbmptnfess	0,079109
3.	subschfess	0,071044
4.	sabdaps	0,018817
5.	zenius_oliv	0,018433

Tabel 4.3 menunjukkan lima pengguna teratas pada metrik *degree centrality* untuk jaringan Zenius. Kelima pengguna tersebut adalah akun schfess, sbmptnfess, subschfess, sabdaps, dan zenius_oliv.

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=False)
    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 simpul di masing-masing jaringan. *Output* dari kode di atas adalah berupa dua *dataframes* yang berisi kumpulan pengguna beserta nilai *betweenness centrality* yang dimiliki pengguna-pengguna tersebut pada masing-masing jaringan. Lima peringkat tertinggi pada *dataframe* untuk jaringan percakapan pengguna Twitter mengenai Ruangguru ditunjukkan pada Tabel 4.4.

Tabel 4.4 Lima Pengguna dengan Nilai *Betweenness Centrality* Tertinggi pada Jaringan Ruangguru

No.	Pengguna	<i>Betweenness Centrality</i>
1.	schfess	1997242,99
2.	subschfess	1349604,86
3.	forskyblue_	1146853,16
4.	hiromi_daiji	1039282,75
5.	sbmptnfess	884745,10

Tabel 4.4 menunjukkan lima pengguna teratas pada metrik *betweenness centrality* untuk jaringan Ruangguru. Kelima pengguna tersebut adalah akun schfess, subschfess, forskyblue,

hiromi_daiji, dan sbmptnfess. Selanjutnya, lima peringkat tertinggi pada *dataframe* untuk jaringan percakapan pengguna Twitter mengenai Zenius ditunjukkan pada Tabel 4.5.

Tabel 4.5 Lima Pengguna dengan Nilai *Betweenness Centrality* Tertinggi pada Jaringan Zenius

No.	Pengguna	<i>Betweenness Centrality</i>
1.	schfess	483501,50
2.	sbmptnfess	260140,35
3.	subschfess	248605,17
4.	zenius_oliv	134208,99
5.	hopefullyperf	101940,44

Tabel 4.5 menunjukkan lima pengguna teratas pada metrik *betweenness centrality* untuk graf Zenius. Kelima pengguna tersebut adalah akun schfess, sbmptnfess, subschfess, zenius_oliv, dan hopefullyperf.

4.3.2.3 Closeness Centrality

Untuk menghitung nilai *closeness centrality* atau rata-rata pada seluruh lintasan terpendek dari suatu simpul ke setiap simpul lainnya di dalam jaringan, penulis menggunakan fungsi *closeness centrality* 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 simpul di masing-masing jaringan. *Output* dari kode di atas adalah berupa dua *dataframes* yang berisi kumpulan pengguna beserta nilai *closeness centrality* yang dimiliki pengguna-pengguna tersebut pada masing-masing jaringan. Lima peringkat tertinggi pada *dataframe* untuk jaringan percakapan pengguna Twitter mengenai Ruangguru ditunjukkan pada Tabel 4.6.

Tabel 4.6 Lima Pengguna dengan Nilai *Closeness Centrality* Tertinggi pada Jaringan Ruangguru

No.	Pengguna	<i>Closeness Centrality</i>
1.	schfess	0,18217
2.	hiromi_daiji	0,17051
3.	subschfess	0,17028
4.	syafiranurainun	0,16526
5.	sbmptnfess	0,16497

Tabel 4.6 menunjukkan lima pengguna teratas pada metrik *closeness centrality* untuk jaringan Ruangguru. Kelima pengguna tersebut adalah akun schfess, hiromi_daiji, subschfess, syafiranurainun, dan sbmptnfess. Selanjutnya, lima peringkat tertinggi pada *dataframe* untuk jaringan percakapan pengguna Twitter mengenai Zenius ditunjukkan pada Tabel 4.7.

Tabel 4.7 Lima Pengguna dengan Nilai *Closeness Centrality* Tertinggi pada Jaringan Zenius

No.	Pengguna	<i>Closeness Centrality</i>
1.	schfess	0,20610
2.	zenius_oliv	0,19737
3.	hopefullyperf	0,19039
4.	byunpov	0,18547
5.	amsterdamlaf	0,18475

Tabel 4.7 menunjukkan lima pengguna teratas pada metrik *closeness centrality* untuk jaringan Zenius. Kelima pengguna tersebut adalah akun schfess, zenius_oliv, hopefullyperf, byunpov, dan amsterdamlaf.

4.3.2.4 Eigenvector Centrality

Untuk menghitung nilai *eigenvector centrality* atau nilai kuantitas dan kualitas koneksi dari seluruh simpul 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

Kode Program 4.19 adalah *script* Python untuk menghitung nilai *eigenvector centrality* pada seluruh simpul di masing-masing jaringan. *Output* dari kode di atas adalah berupa dua *dataframes* yang berisi kumpulan pengguna beserta nilai *eigenvector centrality* yang dimiliki pengguna-pengguna tersebut pada masing-masing jaringan. Lima peringkat tertinggi pada *dataframe* untuk jaringan percakapan pengguna Twitter mengenai Ruangguru ditunjukkan pada Tabel 4.8.

Tabel 4.8 Lima Pengguna dengan Nilai *Eigenvector Centrality* Tertinggi pada Jaringan Ruangguru

No.	Pengguna	<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 4.8 menunjukkan lima pengguna teratas pada metrik *eigenvector centrality* untuk jaringan Ruangguru. Kelima simpul tersebut adalah akun schfess, subschfess, ambisfs, sbmptnfess, dan guidance204. Selanjutnya, lima peringkat tertinggi pada *dataframe* untuk jaringan percakapan pengguna Twitter mengenai Zenius ditunjukkan pada Tabel 4.9.

Tabel 4.9 Lima Pengguna dengan Nilai *Eigenvector Centrality* Tertinggi pada Jaringan Zenius

No.	Pengguna	<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 4.9 menunjukkan lima pengguna teratas pada metrik *eigenvector centrality* untuk jaringan Zenius. Kelima pengguna tersebut adalah akun schfess, subschfess, sbmptnfess, zenius_oliv, dan hopefullyperf.

Tabel 4.10 Seluruh Hasil Perhitungan Metrik *Centrality* pada Jaringan Ruangguru

Pengguna	DC	BC	CC	EC
	<i>Score/Rank</i>	<i>Score/Rank</i>	<i>Score/Rank</i>	<i>Score/Rank</i>
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 4.10 menunjukkan seluruh nilai metrik *centrality* pada jaringan Ruangguru. Dari keempat metrik tersebut, terlihat bahwa pengguna dengan nama akun schfess, subschfess, dan sbmptnfess merupakan pengguna yang selalu menempati lima besar teratas. Hal ini

menunjukkan bahwa ketiga akun tersebut merupakan *key actor* dalam penyebaran informasi pada jaringan Ruangguru.

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

Pengguna	DC	BC	CC	EC
	<i>Score/Rank</i>	<i>Score/Rank</i>	<i>Score/Rank</i>	<i>Score/Rank</i>
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 4.11 menunjukkan seluruh nilai metrik *centrality* pada jaringan Zenius. Dari keempat metrik tersebut, terlihat bahwa pengguna dengan nama akun schfess dan zenius_oliv merupakan pengguna yang selalu menempati lima besar teratas. Hal ini menunjukkan bahwa kedua akun tersebut merupakan *key actor* dalam penyebaran informasi pada jaringan Zenius.

4.4 Visualisasi Model Jaringan

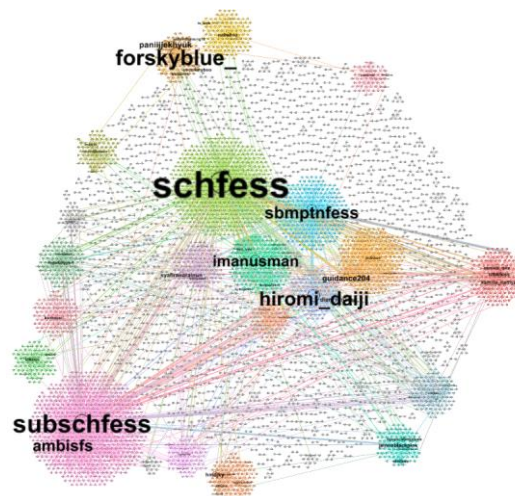
Pada tahapan ini, penulis akan menampilkan representasi visual dari model jaringan yang telah diolah pada tahapan sebelumnya, dengan tujuan agar pembaca lebih mudah untuk mengidentifikasi hasil analisis jaringan. Namun, sebelum itu, penulis akan melakukan tranformasi kedua data graf Networkx ke dalam bentuk *file Graph Exchange XML Format* (GEXF) (Gephi, 2022).

```
# 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 (Gephi, 2022).

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 klaster 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 anggota di atas atau sama dengan 1,00 %.



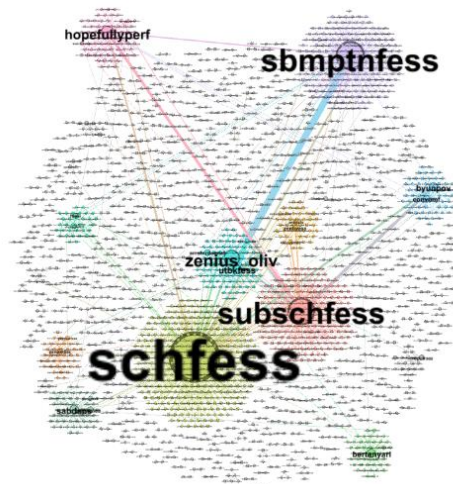
Gambar 4.1 Visualisasi Jaringan Ruangguru

Daftar lima kelompok terbesar pada jaringan Ruangguru beserta atributnya ditunjukkan pada Tabel 4.12.

Tabel 4.12 Lima Kelompok Terbesar pada Jaringan Ruangguru

No .	No. Kelompo k	Jumlah Anggot a	Persentas e	<i>Key Actor</i>
1.	10	581	10,59 %	subschfess
2.	6	452	8,24 %	schfess
3.	4	230	4,19 %	sbmptnfes s
4.	8	204	3,72 %	guidance2 04
5.	0	165	3,01 %	imanusma n

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



Gambar 4.2 Visualisasi Jaringan Zenius

Gambar 4.2 adalah visualisasi model jaringan percakapan pengguna Twitter mengenai Zenius. Terdapat 11 kelompok atau kluster 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 4.13 Lima Kelompok Terbesar pada Jaringan Zenius

No	No. Kelompok	Jumlah Anggota	Persentase	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 dan kelompok 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 simpul di dalam kelompok masing-masing, *key actors* tersebut juga berpengaruh dalam penyebaran informasi pada seluruh simpul di dalam jaringan.

“Halaman ini sengaja dikosongkan”

BAB V

Kesimpulan dan Saran

5.1. Kesimpulan

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

1. Implementasi SNA dalam menganalisis karakteristik suatu jaringan sosial dapat menggunakan metrik *network properties*. Berdasarkan hasil perhitungan dan perbandingan metrik *network properties* yang telah dilakukan pada tahapan sebelumnya, jaringan percakapan pengguna Twitter mengenai perusahaan *edtech* yang paling unggul adalah Zenius dengan total lima dari delapan metrik. Jaringan Zenius unggul dalam metrik *density*, *modularity*, *diameter*, *average path length*, dan *connected components*. Hal ini menunjukkan bahwa pergerakan *trend* jumlah percakapan pengguna Twitter mengenai kedua perusahaan *edtech* tersebut pada 1 Juli 2021 sampai dengan 30 September 2021 tidak berbanding lurus dengan kualitas karakteristik jaringan sosial yang terbentuk.
2. Implementasi SNA dalam mengidentifikasi pengguna Twitter yang paling berpengaruh (*key actor*) di antara semua pengguna lain yang ada di dalam suatu jaringan dapat menggunakan metrik *centrality*. Melalui perhitungan metrik *degree centrality*, *betweenness centrality*, *closeness centrality*, dan *eigenvector centrality* yang telah dilakukan pada tahapan sebelumnya, didapatkan *key actors* pada masing-masing jaringan. *Key actors* pada jaringan Ruangguru adalah akun schfess, subschfess, dan sbmptnfess. Sedangkan, *key actors* pada jaringan Zenius adalah akun schfess dan zenius_oliv. Selain di dalam jaringan, akun-akun tersebut juga merupakan *key actors* 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 pengguna di dalam kelompok, *key actors* tersebut juga berpengaruh dalam penyebaran informasi pada seluruh pengguna di dalam jaringan.

5.2. Saran

Berdasarkan hasil analisis dan penelitian pada tahapan sebelumnya, terdapat beberapa saran bagi perusahaan:

1. Akun Twitter perusahaan dapat lebih aktif lagi melakukan *engagement* atau interaksi dengan *followers* yang dimiliki. Seperti lebih sering mengunggah *tweet* yang menarik atau mengadakan *marketing campaign* yang mengikutsertakan *followers* atau orang lain untuk bergabung. Sehingga percakapan antara pengguna Twitter mengenai perusahaan tersebut, tidak hanya mengalami peningkatan *trend*, tetapi juga dapat membentuk jaringan sosial dengan kualitas karakteristik 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 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 dari situs jejaring sosial lain dengan kata kunci yang lebih variatif dan memperpanjang rentang waktu pengambilan data.
2. Dapat menerapkan metode SNA pada studi kasus yang berbeda dengan menambahkan metrik dari metode-metode lainnya

DAFTAR PUSTAKA

- Antoniadis, I., & Charmantzi, A. (2016). Social network analysis and social capital in marketing: theory and practical implementation. *International journal of technology marketing*, 11(3), 344-359.
- Bhardwaj, R., Yarrow, N., & Cali, M. (2020). *EdTech in Indonesia : Ready for Take-off?* Washington DC.: World Bank.
- Community detection for NetworkX. (2022). *Community detection for NetworkX's documentation*. Retrieved from Community detection for NetworkX: <https://python-louvain.readthedocs.io/en/latest/>. Diakses pada tanggal 18 Januari 2022.
- Fornito, A., Zalesky, A., & Bullmore, E. (2016). *Fundamentals of brain network analysis*. USA: Academic Press.
- Gephi. (2022). *Learn how to use Gephi*. Retrieved from Gephi: <https://gephi.org/users/>. Diakses pada tanggal 19 Januari 2022.
- Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques*. USA: Elsevier.
- Hartsfield, N., & Ringel, G. (2013). *Pearls in graph theory: a comprehensive introduction*. USA: Dover Publication, Inc.
- Himelboim, I., & Golan, G. J. (2019). A social networks approach to viral advertising: The role of primary, contextual, and low influencers. *Social Media+ Society*, 5(3): 1-13.
- Litterio, A. M., Nantes, E. A., Larrosa, J. M., & Gómez, L. J. (2017). Marketing and social networks: a criterion for detecting opinion leaders. *European Journal of Management and Business Economics.*, 26(3): 47-366.
- Mollett, A., Moran, D., & Dunleavy, P. (2011). *Using Twitter in university research, teaching and impact activities*. London: LSE Public Policy Group.
- Network Science. (2021). *Network Science by Albert-László Barabási*. Retrieved from Network Science: <http://networksciencebook.com/> . Diakses pada tanggal 09 November 2022.
- NetworkX . (2022). *Software for Complex Networks — NetworkX 2.6.2 documentation*. Retrieved from NetworkX : <https://networkx.org/documentation/stable/index.html>. Diakses pada tanggal 18 Januari 2022.
- Pandas. (2022). *pandas documentation — pandas 1.3.5 documentation*. Retrieved from Pandas: <https://pandas.pydata.org/docs/>. Diakses pada tanggal 18 Januari 2018.
- Ruangguru. (2021). *Aplikasi Bimbingan Online #1 di Indonesia | Ruangguru*. Retrieved from Ruangguru: <https://www.ruangguru.com/>. Diakses pada tanggal 23 September 2022.
- Zenius. (2021, 23 September). *About - Zenius Education* . Retrieved from Zenius: <https://www.zenius.net/about/>. Diakses pada tanggal 23 September 2022.

“Halaman ini sengaja dikosongkan”

LAMPIRAN

Lampiran 1

Tabel Deskripsi Atribut pada 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 <i>user</i> 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>	<i>User</i> 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 2

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 3

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 4

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 5

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 6

Tabel *Modularity Class* pada Jaringan Ruangguru

Pengguna	<i>Modularity Class</i>
7chillboo	0
kjnchsolo	0
morphoflies	1
junkykime	1
exosalien	1
mactaadid	2
kyutieshii	2
yourarians	3
sbmptnfess	4
mumarisatulhk	5
schfess	6
dian_renataa	3
helloyon4	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 7

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

“Halaman ini sengaja dikosongkan”

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.