



TUGAS AKHIR - KM184801

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

ADRIAN MAULANA MUHAMMAD
061115400000099

Dosen Pembimbing:
Dr. Imam Mukhlash, S.Si, MT
Dr. Darmaji, S.Si, MT

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* MENGGUNAKAN PENDEKATAN
SOCIAL NETWORK ANALYSIS (STUDI KASUS:
RUANGGURU DAN ZENIUS)**

ADRIAN MAULANA MUHAMMAD
NRP 06111540000099

Dosen Pembimbing:
Dr. Imam Mukhlash, S.Si, MT
Dr. Darmaji, S.Si, MT

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

Supervisors:
Dr. Imam Mukhlash, S.Si, MT
Dr. Darmaji, S.Si, MT

DEPARTEMENT 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)

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

TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat
Untuk memperoleh gelar Sarjana Matematika
Pada bidang studi Ilmu Komputer
Program Studi S-1 Departemen Matematika
Fakultas Sains dan Analitika Data
Institut Teknologi Sepuluh Nopember

Oleh:

ADRIAN MAULANA MUHAMMAD
NRP. 06111540000099

Dosen Pembimbing II Menyetujui, Dosen Pembimbing I

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

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

Mengetahui,
Kepala Departemen Matematika
FSAD ITS

Subchan, Ph.D
NIP. 19710513 199702 1 001
Surabaya, 15 Januari 2022

“Halaman ini sengaja dikosongkan”

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

Nama : Adrian Maulana Muhammad
NRP : 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 yang tinggi di Indonesia telah menghasilkan peluang baru bagi bisnis *educational technology (edtech)* dalam mempromosikan produk mereka. Ketersediaan data interaksi pengguna media sosial dapat membantu bisnis *edtech* untuk meningkatkan efektivitas kegiatan pemasaran mereka di media sosial. Penelitian ini mengusulkan implementasi *Social Network Analysis* (SNA) dalam menganalisis interaksi pengguna media sosial mengenai bisnis *edtech*. SNA merupakan pendekatan analitis yang memanfaatkan teori graf untuk mengidentifikasi struktur *network graph*. Metrik SNA yang digunakan pada penelitian ini adalah metrik *network properties* untuk mendeskripsikan model jaringan dan metrik *centrality* untuk mengidentifikasi *key actor* dalam jaringan. Jaringan yang terbentuk berdasarkan percakapan pengguna Twitter mengenai dua perusahaan *edtech* di Indonesia, yaitu Ruangguru dan Zenius.

Kata Kunci: *Social Network Analysis, Twitter, Ruangguru, Zenius*

“Halaman ini sengaja dikosongkan”

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

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

ABSTRACT

The high number of social media users in Indonesia has created new opportunities for educational technology (edtech) businesses to promote their products. The availability of social media user's interaction data can help 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 network graph. SNA metrics used in this study are network properties metrics to describe the network model 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, namely Ruangguru and Zenius.

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

“Halaman ini sengaja dikosongkan”

KATA PENGANTAR

Alhamdulillahirobbil'aalamiin, segala puji syukur bagi 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*
MENGUNAKAN 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 Surabaya yang telah mendukung penulis menyelesaikan kuliahnya.
5. Ibu Dr. Dwi Ratna Sulistyaningrum, 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. Partner penulis yaitu Rochman Dwitok, Safir Takhirizuddin, dan Anindya Rachmawati yang telah mendengarkan keluhan penulis dan mendukung penulis dalam menyelesaikan Tugas Akhir ini.
9. 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, 21 Januari 2022

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN.....	v
ABSTRAK	vii
<i>ABSTRACT</i>	ix
KATA PENGANTAR.....	xi
DAFTAR ISI	xi
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvii
DAFTAR KODE PROGRAM	xix
DAFTAR LAMPIRAN	xxi
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	4
1.3 Batasan Masalah	4
1.4 Tujuan Penelitian	5
1.5 Manfaat Penelitian	5
BAB II TINJAUAN PUSTAKA.....	7
2.1 Penelitian Terdahulu	7
2.2 Ruangguru.....	9
2.3 Zenius.....	9
2.4 Twitter.....	10
2.5 <i>Data Mining</i>	11
2.6 Teori Graf	12
2.7 <i>Social Network Analysis (SNA)</i>	15
2.7.1 Network Properties.....	16
2.7.2 <i>Centrality</i>	22
BAB III METODE PENELITIAN	27
3.1 Objek dan Aspek Penelitian.....	27
3.2 Peralatan Penunjang.....	27
3.3 Langkah-langkah Penelitian.....	28

BAB IV ANALISIS DAN PEMBAHASAN	33
4.1 <i>Data Collection</i>	33
4.2 <i>Data Pre-processing</i>	34
4.2.1 <i>Import Dataset</i>	35
4.2.2 Menghapus <i>Tweet</i> yang Duplikat	35
4.2.3 Menghapus <i>Tweet</i> yang Tidak Mempunyai Interaksi	36
4.2.4 Mengambil <i>Tweet</i> Percakapan Antara Konsumen	37
4.2.5 Pengelompokan <i>Tweet</i> Mengenai Perusahaan <i>Edtech</i>	38
4.2.6 Transformasi Data ke Bentuk <i>Edge List</i>	39
4.3 <i>Social Network Analysis (SNA)</i>	40
4.3.1 <i>Network Properties</i>	41
4.3.2 <i>Centrality</i>	48
4.4 Visualisasi Model Jaringan	58
BAB V KESIMPULAN DAN SARAN	63
5.1. Kesimpulan	63
5.2. Saran	64
DAFTAR PUSTAKA	66
LAMPIRAN	69
BIODATA PENULIS	76

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	3
Gambar 2.1 Graf G	12
Gambar 2.2 <i>3-Regular Graph</i>	13
Gambar 2.3 <i>Path</i> dari <i>Node A</i> ke <i>Node I</i> Pada Graf	13
Gambar 2.4 Graf yang Tidak Terhubung (<i>Disconnected</i>)	14
Gambar 2.5 Graf dengan <i>Weights</i>	14
Gambar 2.6 <i>Directed Graph</i>	15
Gambar 2.7 Ilustrasi Algoritma Louvain	19
Gambar 2.8 Dua <i>nodes</i> berlabel “Bob” yang mempunyai <i>degree centrality</i> yang sama	23
Gambar 3.1 Diagram Alir Penelitian	30
Gambar 3.1 Diagram Alir Penelitian (lanjutan).....	31
Gambar 4.1 Visualisasi Graf Ruangguru	59
Gambar 4.2 Visualisasi Graf Zenius	60

“Halaman ini sengaja dikosongkan”

DAFTAR TABEL

Tabel 2.1 Deskripsi Metrik <i>Network Properties</i>	21
Tabel 2.2 Deskripsi Metrik <i>Centrality</i>	26
Tabel 3.1 Spesifikasi Perangkat.....	26
Tabel 3.2 Aplikasi dan Teknologi	26
Tabel 4.1 <i>Network Properties</i> pada Graf Ruangguru dan Zenius	26
Tabel 4.2 <i>Degree Centrality</i> Tertinggi pada Graf Ruangguru....	26
Tabel 4.3 <i>Degree Centrality</i> Tertinggi pada Graf Zenius.....	26
Tabel 4.4 <i>Betweenness Centrality</i> Tertinggi pada Graf Ruangguru	26
Tabel 4.5 <i>Betweenness Centrality</i> Tertinggi pada Graf Zenius ..	26
Tabel 4.6 <i>Closeness Centrality</i> Tertinggi pada Graf Ruangguru	26
Tabel 4.7 <i>Closeness Centrality</i> Tertinggi pada Graf Zenius	26
Tabel 4.8 <i>Eigenvector Centrality</i> Tertinggi pada Graf Ruangguru	26
Tabel 4.9 <i>Eigenvector Centrality</i> Tertinggi pada Graf Zenius ...	26
Tabel 4.10 Seluruh Metrik <i>Centrality</i> pada Graf Ruangguru	26
Tabel 4.11 Seluruh Metrik <i>Centrality</i> pada Graf Zenius	26
Tabel 4.12 Lima Kelompok Terbesar pada Graf Ruangguru	60
Tabel 4.13 Lima Kelompok Terbesar pada Graf Zenius	61

“Halaman ini sengaja dikosongkan”

DAFTAR KODE PROGRAM

Kode Program 4.1 <i>Scraping</i> Data Twitter	34
Kode Program 4.2 <i>Import Dataset</i>	35
Kode Program 4.3 Menghapus <i>Tweet</i> yang Duplikat.....	36
Kode Program 4.4 Menghapus <i>Tweet</i> yang Tidak Mempunyai Interaksi	36
Kode Program 4.5 Mengambil <i>Tweet</i> Percakapan Antara Konsumen	37
Kode Program 4.6 Pengelompokan <i>Tweet</i> Mengenai Perusahaan <i>Edtech</i>	38
Kode Program 4.7 Transformasi Data ke Bentuk <i>Edge List</i>	39
Kode Program 4.8 Transformasi <i>Edge List</i> ke <i>Undirected Graph</i>	40
Kode Program 4.9 <i>Order</i> dan <i>Size</i> dari Graf	41
Kode Program 4.10 <i>Density</i> dari Graf.....	42
Kode Program 4.11 <i>Modularity</i> pada Graf.....	43
Kode Program 4.12 <i>Diameter</i> pada Graf.....	44
Kode Program 4.13 <i>Average Path Length</i> pada Graf.....	45
Kode Program 4.14 <i>Average Degree</i> pada Graf.....	46
Kode Program 4.15 <i>Connected Components</i> pada Graf.....	47
Kode Program 4.16 <i>Degree Centrality</i> pada Graf.....	49
Kode Program 4.17 <i>Betweenness Centrality</i> pada Graf	51
Kode Program 4.18 <i>Closeness Centrality</i> pada Graf.....	53
Kode Program 4.19 <i>Eigenvector Centrality</i> pada Graf	55
Kode Program 4.20 Transformasi Data Graf Networkx ke <i>File</i> <i>GEXF</i>	55

“Halaman ini sengaja dikosongkan”

DAFTAR LAMPIRAN

Lampiran 1	69
Lampiran 2	70
Lampiran 3	71
Lampiran 4	72
Lampiran 5	73
Lampiran 6	74
Lampiran 7	75

“Halaman ini sengaja dikosongkan”

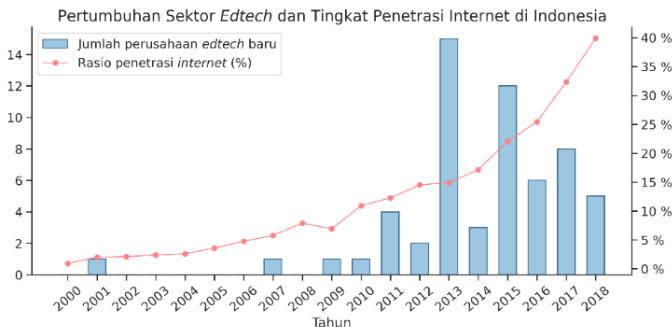
BAB I

PENDAHULUAN

Pada bagian ini, penulis membahas tentang latar belakang yang mendasari penulisan penelitian. Kemudian penulis membahas rumusan masalah, batasan masalah, tujuan, dan manfaat yang diambil berdasarkan latar belakang penyusunan penelitian ini.

1.1 Latar Belakang

Indonesia merupakan salah satu negara yang mempunyai jumlah pengguna internet yang tinggi. Penggunaan internet yang tinggi di Indonesia berpengaruh terhadap munculnya inovasi pada penggunaan informasi, komunikasi, dan teknologi untuk pendidikan dalam bentuk platform *educational technology* (*edtech*). Sektor perusahaan *edtech* di Indonesia masih dalam fase pertumbuhan, dengan hampir semua pemain utama masih terus melakukan eksperimen pada produk ataupun pasar mereka. Pada Gambar 1.1 menunjukkan bahwa peningkatan pada pendirian perusahaan *edtech* juga bertepatan dengan peningkatan penetrasi internet di Indonesia [1].



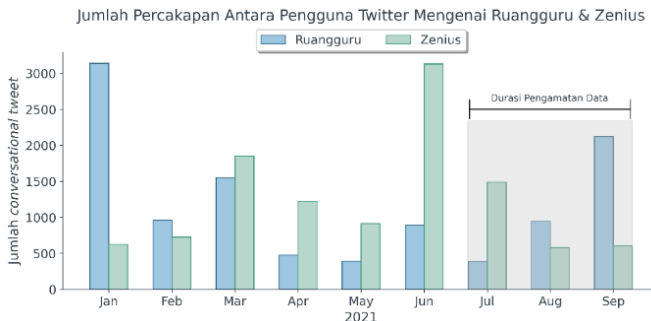
Gambar 1.1 Pertumbuhan Sektor *Edtech* dan Tingkat Penetrasi Internet di Indonesia
(Sumber Data: World Bank Tahun 2020 [1])

Dua perusahaan *edtech*, yaitu Ruangguru dan Zenius, termasuk dalam daftar perusahaan *edtech* yang menonjol dalam hal pertumbuhan *user* dan perhatian investor selama beberapa tahun terakhir [1]. Selain hal tersebut, pada September 2021, Ruangguru dan Zenius juga termasuk akun platform *edtech* di Indonesia yang mempunyai pengikut terbanyak di situs jejaring sosial *online* Twitter. Namun, dengan memiliki jumlah *follower* Twitter yang tinggi, tidak berarti banyak jika tingkat interaksi rendah dan perusahaan tidak melibatkan *follower* 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 *node* (simpul), dan interaksi antara pengguna Twitter tersebut, dilambangkan dengan *edge* (sisi).

Sekitar awal Juli sampai dengan akhir September 2021, terlihat bahwa jumlah percakapan pengguna Twitter mengenai Zenius menurun, sedangkan pada Ruangguru 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, 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 terhadap penyebaran informasi pada jaringan sosial tersebut. Pada SNA, *key actor* di suatu jaringan sosial dapat dianalisis dengan metrik *centrality*. Durasi pengamatan data tersebut 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 pada 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 [2]. Kedua, penelitian yang ditulis oleh Arnaldo Litterio dkk. mengenai penerapan SNA dalam pemasaran untuk mengidentifikasi *opinion leaders* [3]. Ketiga, penelitian yang ditulis oleh Itai Himelboim dan Guy Golan mengenai pendekatan *social networks* dalam menganalisis peran *influencer* pada *viral advertising* [4].

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 *nodes* dan 4.982 *edges* untuk Ruangguru dan 2.605 *nodes* dan 2.123 *edges* untuk Zenius.

Berdasarkan latar belakang yang telah diuraikan di atas, maka penulis melakukan suatu penelitian tentang analisis interaksi pengguna Twitter mengenai dua perusahaan *edtech*, 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 percakapan pengguna Twitter mengenai Ruangguru dan Zenius?
2. Bagaimana implementasi SNA dalam menganalisis pengguna Twitter yang paling berpengaruh (*key actor*) dalam penyebaran informasi mengenai Ruangguru dan Zenius di Twitter?

1.3 Batasan Masalah

Penulis membatasi permasalahan pada penelitian ini sebagai berikut:

1. Data yang akan 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 akan dianalisis adalah percakapan atau *reply* yang merupakan

suatu tanggapan atas suatu *tweet* yang diberikan untuk pembuat *tweet*.

3. Jenis graf yang digunakan adalah *undirected graph*.
4. Proses pengumpulan data dilakukan dengan metode *scraping* pada *website* Twitter yang dapat bebas diakses 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. Menganalisis pengguna Twitter yang paling berpengaruh (*key actor*) dalam penyebaran informasi mengenai Ruangguru dan Zenius di Twitter dengan menggunakan metrik *centrality*.

1.5 Manfaat Penelitian

Adapun manfaat yang diharapkan dari penelitian ini adalah:

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. Serta untuk mengetahui perhitungan dan analisis metrik *network properties* dan *centrality* pada SNA.

“Halaman ini sengaja dikosongkan”

BAB II

TINJAUAN PUSTAKA

Pada bagian ini dijelaskan beberapa teori dasar yang mendukung dalam pengerjaan penelitian ini yang meliputi penelitian terdahulu, penjelasan tentang Ruangguru, Zenius, Twitter, *data mining*, teori graf, dan SNA.

2.1 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 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 salah satu 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 [2].

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 klasifikasi individu dalam jaringan sosial yang terbentuk adalah dengan kombinasi dari metrik *eigenvector centrality* dan *betweenness centrality*. Studi kasus pada penelitian ini adalah suatu situs halaman Facebook suatu 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 [3].

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 *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* [4].

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 *network graph* yang dianalisis berdasarkan percakapan di Twitter mengenai dua perusahaan *edtech*, yaitu Ruangguru dan Zenius.

2.2 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% 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 merupakan salah satu *startup* terbesar pada sektor *edtech* di Indonesia dengan pendapatan 4,4 juta USD pada tahun 2018. Selama pandemi COVID-19, Ruangguru mengalami lonjakan pengguna sebesar 46% menjadi 22 juta akun sepanjang tahun 2020. [5].

2.3 Zenius

Zenius adalah pelopor *startup* teknologi pendidikan di Indonesia yang menggunakan metode *adaptive learning*. Zenius

memproduksi konten pendidikan untuk semua level pendidikan dari SD, SMP, dan SMA beserta persiapan ujian nasional dan tes masuk perguruan tinggi. Zenius memulai perjalanannya pada tahun 2008 dengan produk CD/DVD, sebelum beralih ke *website* pada tahun 2010. Hingga sekarang, Zenius berhasil meluncurkan *mobile apps* yang bisa diunggah melalui iOS dan Android. Zenius telah memiliki lebih dari 74,000 video pembelajaran dan puluhan ribu latihan soal. Pada tahun 2020, Zenius melihat pendapatannya meningkat lebih dari 70% dibandingkan dengan periode yang sama di tahun 2019. Platform ini juga diklaim memiliki konten pendidikan terbesar di Indonesia dengan lebih dari 90.000 video untuk siswa SD hingga SMA [6].

2.4 Twitter

Twitter adalah layanan 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 si pembuat *tweet* tersebut, *retweet* adalah mempublikasikan kembali suatu *tweet* dari pengguna, *mention* merupakan *tweet* yang berisi nama pengguna, *hashtag* digunakan untuk mengindeks 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 [7].

2.5 Data Mining

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

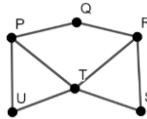
Langkah 1 hingga 4 adalah berbagai bentuk dari *data pre-processing*, di mana mempersiapkan data sebelum proses *mining*. Pola yang menarik disajikan kepada pengguna dan dapat disimpan sebagai suatu *insights* atau pengetahuan. Oleh karena itu, *data mining* adalah proses menemukan pola yang menarik 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 [8].

2.6 Teori Graf

Teori graf berkaitan dengan berbagai jenis jaringan, atau bisa dikatakan, merupakan pemodelan dari suatu jaringan yang disebut graf. Graf yang dimaksud di sini sering digambarkan sebagai simpul (*vertex/node*) yang dihubungkan oleh sisi (*edge*). Suatu Graf G adalah pasangan himpunan (V, E) di mana elemen dari himpunan V adalah *nodes* dan elemen dari himpunan E adalah *edges* dari Graf G . Himpunan V pada graf G bisa dinotasikan dengan $V(G)$ dan himpunan E pada Graf G dinotasikan dengan $E(G)$ [9]. Sehingga, Graf G dapat direpresentasikan sebagai berikut:

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

Berikut adalah salah satu contoh dari graf:



Gambar 2.1 Graf G

(Sumber: Dokumen Penulis)

Pada Gambar 2.1 terdapat Graf G yang memiliki 6 *nodes* dan 8 *edges*. Adapun himpunan V dan E dari Graf G adalah sebagai berikut:

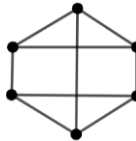
$$V(G) = \{P, Q, R, S, T, U\} \quad (2.2)$$

$$E(G) = \{(P, T), (P, Q), (P, U), (Q, R), (T, R), (R, S), (T, S), (U, T)\} \quad (2.3)$$

Selain *nodes* dan *edges*, berikut beberapa terminologi dasar lainnya dalam teori graf yang digunakan pada penelitian ini [9]:

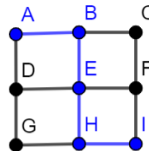
1. *Degree* (derajat) pada suatu *node* adalah banyaknya *edges* pada *node* tersebut. Suatu *node* dengan *degree* 0 disebut

isolated node. Pada Gambar 2.2, seluruh *nodes* pada graf tersebut mempunyai *degree* yang sama, graf dengan kriteria itu disebut dengan *regular graph*. Nilai *degree* pada masing-masing *node* dari *regular graph* tersebut adalah 3.



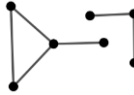
Gambar 2.2 3-Regular Graph
(Sumber: Dokumen Penulis)

2. *Path* (lintasan) adalah barisan *nodes* yang menggunakan *edges*, dari suatu *node* ke *node* yang dituju (antara dua *nodes*). *Path* dari *node A* ke *node I* ditunjukkan dengan *edges* berwarna biru pada Gambar 2.3, *path* tersebut adalah salah satu dari banyak kemungkinan *paths* yang ada pada graf.



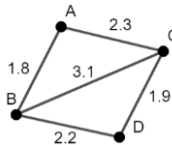
Gambar 2.3 *Path* dari *Node A* ke *Node I* Pada Graf
(Sumber: Dokumen Penulis)

3. Suatu graf dikatakan *connected* (terhubung) jika terdapat *path* dari sembarang *nodes* ke *nodes* lainnya. Pada Gambar 2.4 adalah contoh graf yang *disconnected* (tidak terhubung).



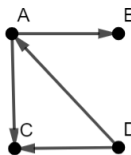
Gambar 2.4 Graf yang Tidak Terhubung (*Disconnected*)
(Sumber: Dokumen Penulis)

4. Tergantung pada masalah yang dianalisis, terkadang *weights* (bobot) diberikan pada *edges* dari suatu graf. *Weights* dapat mewakili jarak antara dua lokasi, waktu tempuh, atau biaya perjalanan. Penting untuk dicatat bahwa jarak antara *nodes* dalam graf tidak selalu sesuai dengan *weights* yang diberikan pada *edges*.



Gambar 2.5 Graf dengan *Weights*
(Sumber: Dokumen Penulis)

5. Berdasarkan orientasi arah pada *edges*, maka graf dibedakan menjadi dua jenis. Pertama, *undirected graph* (graf tak berarah). Pada *undirected graph*, tidak memperhatikan urutan pasangan *nodes* yang dihubungkan oleh *edges*, dengan demikian pada himpunan E berlaku $(V_j, V_k) = (V_k, V_j)$. Setiap graf yang telah dibahas sebelumnya merupakan contoh dari *undirected graph*. Kedua, *directed graph* (graf berarah). *Directed graph* adalah graf yang pada setiap *edges* diberikan orientasi arah. Pada jenis graf ini, himpunan E berlaku $(V_j, V_k) \neq (V_k, V_j)$. Contoh *directed graph* ditunjukkan pada Gambar 2.6.



Gambar 2.6 *Directed Graph*
(Sumber: Dokumen Penulis)

2.7 Social Network Analysis (SNA)

Social Network Analysis (SNA) merupakan pendekatan analitis yang dapat mengidentifikasi struktur jaringan sosial dengan memanfaatkan teori graf. Jaringan sosial yang terbentuk dapat terdiri dari *user*, dilambangkan dengan *node*, dan interaksi antara *user*, dilambangkan dengan *edge*. SNA diperlukan karena membawa kesempatan baru untuk memahami individu atau kelompok terkait pola interaksi mereka. SNA telah digunakan dalam penelitian sosial sejak tahun 1930-an. Dalam penggunaan sebelumnya, penelitian SNA menggunakan metode wawancara dan observasi untuk mendapatkan informasi tentang kualitas interaksi sosial pada suatu komunitas. Untuk menghindari kerumitan, penelitian SNA terdahulu biasanya dilakukan dalam komunitas terbatas [9].

Perkembangan teknologi komputasi saat ini telah memungkinkan peneliti untuk memproses kumpulan data jaringan dengan ukuran yang besar. Ribuan atau bahkan ratusan ribu *nodes* dan *edges* dapat diproses dengan menggunakan komputasi paralel. Tetapi ketika ukuran jaringan terlalu besar, menganalisis grafik dengan menggunakan visualisasi saja sering dianggap tidak efektif. Oleh karena itu, dibutuhkan suatu metrik SNA yang dapat menggambarkan struktur jaringan yang terbentuk. SNA menggunakan pendekatan dari teori graf untuk menyediakan *properties* yang nantinya dapat digunakan sebagai metrik. Pada

penelitian ini, penulis membagi metrik menjadi dua jenis, yaitu *network properties* dan *centrality* [9].

2.7.1 Network Properties

SNA memiliki beberapa metrik *network properties* yang 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*.

2.7.1.1 Order dan Size

Pada suatu jaringan, *order* adalah banyaknya *nodes*, *size* adalah banyaknya *edges* pada jaringan tersebut. Banyaknya *order* dan *size* dalam suatu jaringan sosial menunjukkan banyaknya *user* yang berinteraksi. Dalam pembahasan berikutnya, banyaknya *nodes* akan direpresentasikan sebagai variabel n dan banyaknya *edges* direpresentasikan sebagai variabel m [9].

2.7.1.2 Density

Density pada suatu jaringan merupakan ukuran seberapa banyak *edges* yang ada dibandingkan dengan seberapa banyak *edges* maksimum yang mungkin ada di jaringan tersebut. *Density* menggambarkan kerapatan pada jaringan, semakin tinggi nilai *density* maka semakin baik karena menggambarkan bahwa *nodes* yang ada dalam suatu jaringan saling terhubung satu sama lain. Formula untuk menghitung *density* adalah sebagai berikut [9]:

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

Keterangan:

m : jumlah *edges* pada jaringan

n : jumlah *nodes* pada jaringan

2.7.1.3 Modularity

Modularity merupakan metrik yang digunakan untuk mengetahui kualitas pembagian jaringan ke dalam kelompok atau klaster. Formula untuk mendapatkan nilai *modularity* adalah sebagai berikut [9]:

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

Keterangan:

- $A_{i,j}$: matriks ketetanggaan antara *node i* dan *node j*
- k_i : jumlah *edges* yang melekat (*degree*) pada *node i*
- m : jumlah *edges* pada jaringan
- c_i : kelompok *i*
- $\delta(c_i, c_j)$: Kronecker Delta

Untuk memaksimalkan nilai *modularity* secara efisien dapat dilakukan dengan algoritma Louvain. Algoritma Louvain adalah suatu algoritma *community detection* yang secara rekursif menggabungkan kelompok menjadi satu *node* dan mengeksekusi *modularity clustering* pada jaringan tersebut yang telah diringkas. Algoritma Louvain terdiri dari dua tahapan yaitu *Modularity Optimization* dan *Community Aggregation* [9].

1. *Modularity Optimization*

Pertama, tahapan ini secara acak mengurutkan seluruh *nodes* pada jaringan. Kemudian, satu per-satu *node* dipindahkan ke kelompok yang berbeda, dimisalkan kelompok *C*. *Node* akan terus dipindahkan ke kelompok lain jika terdapat peningkatan nilai modularitas, jika tidak terdapat peningkatan nilai modularitas, maka *node* tersebut tetap berada di kelompok aslinya. Proses ini terus diterapkan ke seluruh *nodes* sampai tidak ada peningkatan yang signifikan pada nilai modularitas. Perubahan nilai

modularitas (ΔM) dapat dihitung dengan formula sebagai berikut [9]:

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

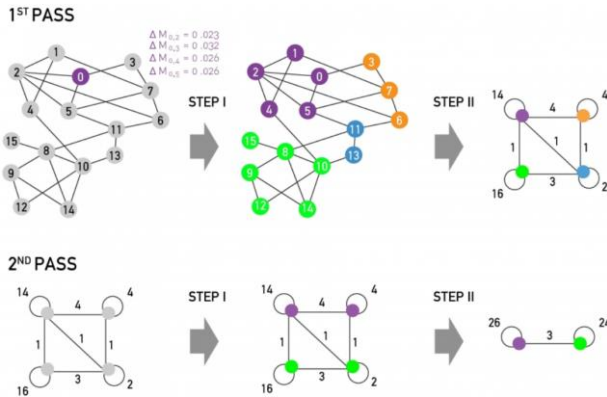
Keterangan:

- \sum_{in} : jumlah *edges* di dalam kelompok *C*
- \sum_{tot} : jumlah *edges* yang melekat dengan *nodes* yang ada di kelompok *C*
- k_i : jumlah *edges* yang melekat pada *node i*
- $k_{i,in}$: jumlah *edges* pada *node i* yang melekat pada kelompok *C*
- m : jumlah keseluruhan *edges* dalam jaringan

2. *Community Aggregation*

Setelah menyelesaikan tahapan pertama, semua *nodes* pada kelompok yang sama akan digabung menjadi satu *node* raksasa. Bobot pada *node* raksasa tersebut adalah jumlah dari seluruh bobot *nodes* yang ada pada kelompok. Tahapan ini menghasilkan *output* berupa *nodes* yang *self-loop* yang di mana merupakan jumlah dari semua *nodes* yang ada pada kelompok, dengan kata lain, *nodes* yang diringkas menjadi hanya satu *node* raksasa [9].

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 tercapai. Ilustrasi algoritma Louvain ditunjukkan pada Gambar 2.7 [9].



Gambar 2.7 Ilustrasi Algoritma Louvain
(Sumber: Barabási, Albert-László Tahun 2016 [9])

2.7.1.4 Diameter

Diameter adalah *shortest path* terpanjang atau terjauh antara sepasang *nodes* dalam suatu jaringan. Semakin kecil nilai *diameter* maka semakin baik, karena proses penyebaran informasi antara suatu *node* dengan *node* lainnya, dengan jarak yang terjauh, hanya perlu melewati sedikit *nodes*. Nilai *diameter* yang kecil mengindikasikan proses penyebaran informasi akan memakan waktu lebih sedikit karena melibatkan lebih sedikit aktor. Untuk jaringan yang berukuran besar, *shortest path* dapat ditentukan menggunakan algoritma *Breadth-First Search* (BFS). *Diameter* dari suatu jaringan biasa dinotasikan sebagai d_{max} . Identifikasi *shortest path* antara *node i* dan *node j* dengan algoritma BFS mengikuti langkah-langkah berikut [9]:

1. Mulai dari *node i* yang kita beri label “0”.
2. Temukan *node* yang terhubung langsung ke *i*. Beri label “1” dan tempatkan di dalam antrian.

3. Ambil *node* pertama yang berlabel n keluar dari antrian ($n = 1$ sebagai langkah pertama). Temukan *node* yang belum berlabel dan berdekatan dengan *node* yang dipilih, lalu beri label “ $n+1$ ” dan masukkan dalam antrian.
4. Ulangi langkah 3 sampai anda menemukan *node* j yang merupakan target *node* atau tidak ada lagi *node* dalam antrian.
5. Jarak antar i dan j adalah label untuk j . Jika j tidak mempunyai label, maka $d_{ij} = \infty$.

2.7.1.5 Average Path Length

Average path length merupakan rata-rata *shortest path* di antara setiap pasang *nodes* yang ada di dalam suatu jaringan. Semakin kecil nilai *average path length*, maka semakin baik, karena artinya rata-rata jarak yang harus ditempuh untuk melakukan penyebaran informasi adalah lebih pendek. *Average path length* biasa dinotasikan sebagai $\langle d \rangle$. Kita dapat menggunakan algoritma BFS, seperti yang sudah dibahas sebelumnya, untuk menentukan bobot *shortest path* pada jaringan yang besar. Formula untuk menghitung *average path length* adalah sebagai berikut [9].

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

Keterangan:

n : jumlah *nodes* pada jaringan

$d_{i,j}$: bobot *shortest path* antara *node* i dan *node* j

2.7.1.6 Average Degree

Average degree merupakan rata-rata dari *degree* atau jumlah *edges* yang menghubungkan suatu *node* ke *nodes* lain pada suatu jaringan. Semakin besar nilai *average degree* yang dimiliki

oleh jaringan maka semakin baik, karena apabila suatu *node* menyebarkan informasi ke banyak *nodes*, secara langsung mempercepat penyebaran informasi. Formula untuk menghitung *average degree* adalah sebagai berikut [9]:

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

Keterangan:

- n : jumlah *nodes* pada jaringan
 k_i : *degree* pada *node* ke- i

2.7.1.7 *Connected Components*

Connected components adalah bagian-bagian yang terpisah atau tidak terkoneksi dengan keseluruhan jaringan. Semakin kecil nilai *connected components* maka semakin baik, karena artinya *nodes* tidak terlalu banyak terpisah dalam grup-grup kecil yang tidak saling terkoneksi. *Connected components* dapat dicari dengan menggunakan algoritma BFS yang telah dibahas sebelumnya, karena urutan dari *node* tidak menjadi masalah. Deskripsi keseluruhan metrik *network properties* yang digunakan pada penelitian ini ditunjukkan pada Tabel 2.1 [9].

Tabel 2.1 Deskripsi Metrik *Network Properties*

No.	<i>Network Properties</i>	Deskripsi
1.	<i>Order</i>	Banyaknya <i>nodes</i> di jaringan.
2.	<i>Size</i>	Banyaknya <i>edges</i> 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> pada sepasang <i>nodes</i> terjauh di jaringan.
6.	<i>Average Path Length</i>	Rata-rata <i>shortest path</i> pada setiap pasang <i>nodes</i> di jaringan.
7.	<i>Average Degree</i>	Rata-rata <i>degree</i> pada setiap <i>nodes</i> di jaringan.
8.	<i>Connected Components</i>	Kelompok <i>nodes</i> yang terpisah atau tidak terkoneksi dengan keseluruhan jaringan.

(Sumber: Barabási, Albert-László Tahun 2016 [9])

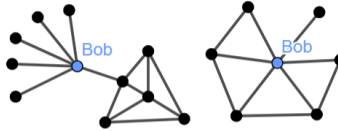
2.7.2 Centrality

Setelah metrik *network properties*, metrik selanjutnya adalah *centrality*. Pengukuran pada *centrality* bertujuan untuk mengidentifikasi *node* yang merupakan pusat penyebaran informasi di antara semua *nodes* yang ada dalam suatu jaringan. Terdapat empat pengukuran *centrality* pada penelitian ini, yaitu: *degree centrality*, *betweenness centrality*, *closeness centrality*, dan *eigenvector centrality* [10].

2.7.2.1 Degree Centrality

Degree centrality menggambarkan ukuran *social connections* yang dimiliki *node* dalam jaringan. *Node* dengan nilai *degree centrality* tertinggi bisa saja merupakan inti dari jaringan, namun bisa saja berada jauh di tepi jaringan. Sebagai contoh, pada Gambar 2.8 menunjukkan bahwa kedua *nodes* yang berlabel “Bob” memiliki nilai *degree centrality* yang sama, akan tetapi peran yang mereka mainkan sangat berbeda. *Node* berlabel “Bob” yang berada

di sebelah kanan sangat sentral, sedangkan yang di sebelah kiri berada di tepi. Hal ini menunjukkan, walaupun *degree centrality* dapat mengidentifikasi *node* yang paling banyak memiliki *social connections*, tetapi metrik ini tidak selalu menunjukkan *node* mana yang berada di tengah jaringan [10].



Gambar 2.8 Dua nodes berlabel “Bob” yang mempunyai *degree centrality* yang sama (Sumber: Dokumen Penulis)

Berikut adalah formula untuk *degree centrality* untuk *node i* [10]:

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

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

Keterangan:

n : jumlah *nodes* pada jaringan

2.7.2.2 Betweenness Centrality

Gagasan singkat pada *betweenness centrality* adalah suatu metrik *centrality* yang tidak mementingkan seberapa banyak *edges* yang dimiliki suatu *node*, tetapi lebih di mana *node* tersebut ditempatkan pada jaringan. Sebagai contoh, pada Gambar 2.8, *node* dengan label “Bob” yang berada di kiri merupakan *node* yang memungkinkan untuk informasi dapat diteruskan dari kelompok kanan ke kelompok kiri, dan sebaliknya. Dengan demikian, *node* berlabel “Bob” tersebut sangat penting bagi arus informasi pada jaringan. Hal ini yang berusaha ditangkap oleh perhitungan *betweenness centrality*. Untuk menghitung *betweenness centrality*

pada suatu *node i*, kita menghitung proporsi *shortest paths* antara *node j* dan *h* yang melewati *node i* [10].

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

Keterangan:

- $\rho_{hj}(i)$: jumlah *shortest paths* antara *node h* dan *node j* yang melalui *node i*
- ρ_{hj} : jumlah *shortest paths* antara *node h* dan *node j*
- n : jumlah *nodes* pada jaringan

2.7.2.3 Closeness Centrality

Closeness centrality adalah mencari *node* yang paling dekat dengan semua *nodes* lainnya. *Closeness centrality* untuk suatu *node* adalah invers dari rata-rata jarak *shortest paths* dari *node* tersebut ke setiap *nodes* lain dalam jaringan. Formula untuk menghitung *closeness centrality* pada *node i* adalah sebagai berikut [10]:

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

Keterangan:

- l_{ij} : bobot *shortest path* antara *node i* dan *node j*
- n : jumlah *nodes* pada jaringan

2.7.2.4 Eigenvector Centrality

Eigenvector centrality adalah ukuran yang memperhitungkan kuantitas dan kualitas koneksi suatu *node*, hal tersebut mempertimbangkan baik derajat dari *node* tersebut dan juga derajat dari *nodes* yang terhubung dengannya. Suatu *node* mungkin memiliki nilai *degree*, *closeness*, atau *betweenness centrality* yang rendah, tetapi *node* tersebut masih bisa mempunyai pengaruh. Secara intuitif, ukuran ini mempertimbangkan tidak

hanya berapa banyak orang yang dikenal, tetapi juga siapa yang dikenal [10].

Untuk memperkirakan ukuran ini, kita harus mempertimbangkan *eigenvalues* dan *eigenvectors* dari matriks ketetanggaan. Untuk mencari *eigenvalues* maka menggunakan persamaan karakteristik polinomial sebagai berikut [10]:

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

Keterangan:

A : matriks ketetanggaan berukuran $n \times n$

λ : *eigenvalues*

I : matriks identitas

Selanjutnya, yaitu mencari *eigenvector* dengan menggunakan *eigenvalue* terbesar, sebagai berikut [10]:

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

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

Keterangan:

A : matriks ketetanggaan berukuran $n \times n$

λ : *eigenvalue* terbesar

\vec{v} : *eigenvector*

I : matriks identitas

Notasi \vec{v} merupakan *eigenvector* matriks $n \times 1$ yang dapat direpresentasikan sebagai persamaan berikut [10]:

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

Eigenvector centrality dari *node i* dapat didefinisikan sebagai *input* ke-*i* dalam *eigenvector* \vec{v} dengan menggunakan *eigenvalue* terbesar dari matriks ketetanggaan *A*. Untuk menormalisasi nilai *eigenvector centrality* dari suatu *node* dapat dihitung dengan cara membagi seluruh nilai *eigenvector* \vec{v} dengan nilai yang tertinggi. Deskripsi keseluruhan metrik *centrality* yang digunakan pada penelitian ini ditunjukkan pada Tabel 2.2 [10].

Tabel 2.2 Deskripsi Metrik Centrality

No.	Centrality	Deskripsi
1.	<i>Degree Centrality</i>	<i>Node</i> dengan jumlah <i>edges</i> terbanyak.
2.	<i>Betweenness Centrality</i>	<i>Node</i> yang terletak di antara <i>nodes</i> lainnya.
3.	<i>Closeness Centrality</i>	<i>Node</i> yang paling dekat dengan <i>nodes</i> lainnya.
4.	<i>Eigenvector Centrality</i>	<i>Node</i> dengan kualitas koneksi terbaik.

(Sumber: Fornito, Alex dkk. Tahun 2016 [10])

BAB III

METODE PENELITIAN

Pada bagian ini akan dibahas mengenai objek dan aspek penelitian, peralatan penunjang, dan langkah-langkah penelitian.

3.1 Objek dan Aspek Penelitian

Objek dari penelitian ini adalah *tweets* di situs jejaring sosial Twitter mengenai dua perusahaan *edtech*, yaitu Ruangguru dan Zenius, pada rentang waktu 1 Juli 2021 sampai dengan 30 September 2021. Metode dalam penelitian ini merupakan metode kuantitatif karena merupakan penelitian yang dilakukan dengan melibatkan investigasi sistematis terhadap suatu fenomena dengan mengumpulkan data yang dapat diukur dan melakukan teknik statistik, matematika atau komputasi. Berdasarkan tujuan penelitian, penelitian ini termasuk dalam penelitian deskriptif, karena bertujuan untuk mendeskripsikan karakteristik dari seseorang, kegiatan atau situasi.

3.2 Peralatan Penunjang

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

Nama Perangkat	Lenovo IdeaPad S340
Processor	AMD Ryzen 3 3200U Radeon Vega Mobile Gfx 2.60 GHz
RAM	8,00 GB

Sistem Operasi	Windows 11
Tipe Sistem	64-bit <i>operating system</i> , x64-based <i>processor</i>

(Sumber: Dokumen Penulis)

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
<i>Code Editor</i>	Visual Studio Code 1.63
Bahasa Pemrograman	Python 3.8.10
<i>Library</i>	<ul style="list-style-type: none"> • Pandas • Twint • Asyncio • Networkx • Community

(Sumber: Dokumen Penulis)

3.3 Langkah-langkah Penelitian

Langkah-langkah sistematis yang akan dilakukan dalam proses pengerjaan penelitian ini adalah sebagai berikut:

1. Studi Literatur

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

2. Identifikasi Permasalahan

Identifikasi permasalahan adalah 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 (*Twitter Intelligence Tool*). Data yang akan dilakukan *scraping* 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*

Analisis *centrality* dilakukan dengan menggunakan *library* Networkx dari bahasa pemrograman Python untuk melihat nilai dari masing-masing metrik *centrality* pada setiap aktor dalam jaringan. Perhitungan *centrality* yang

dilakukan yaitu: *degree centrality*, *betweenness centrality*, *closeness centrality*, dan *eigenvector centrality*.

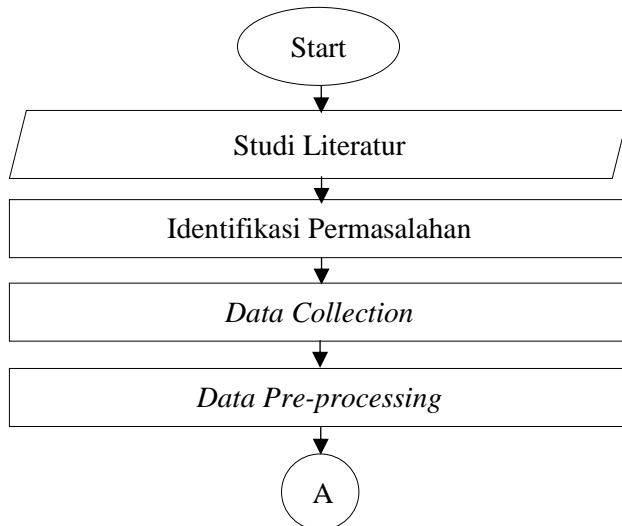
7. Visualiasi Model Jaringan

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

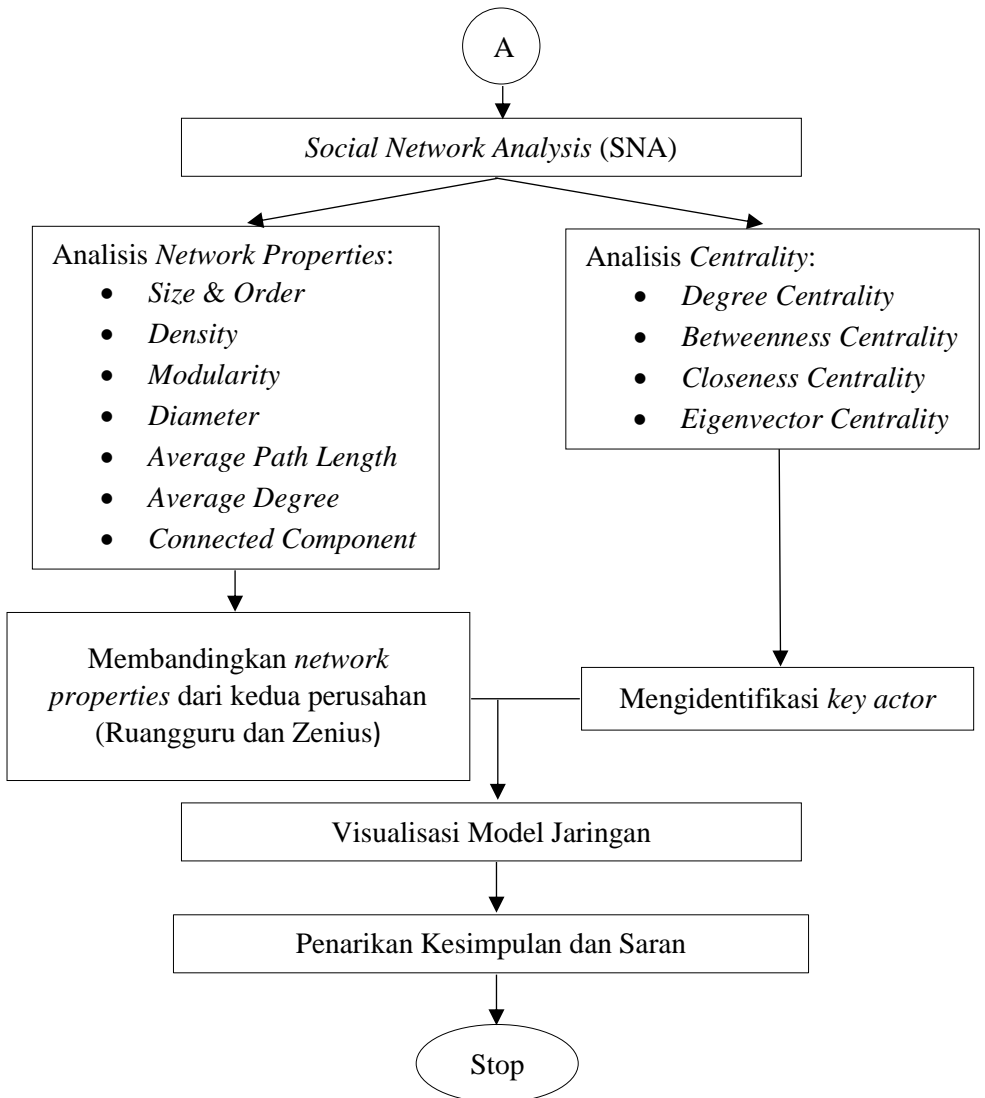
8. Kesimpulan dan Saran

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
(Sumber: Dokumen Penulis)



Gambar 3.2 Diagram Alir Penelitian (lanjutan)
(Sumber: Dokumen Penulis)

“Halaman ini sengaja dikosongkan”

BAB IV

ANALISIS DAN PEMBAHASAN

Bab ini menguraikan dan menjelaskan mengenai beberapa tahapan analisis dalam penelitian ini, seperti *data collection*, *data pre-processing*, implementasi *Social Network Analysis* (SNA) pada jaringan, dan visualisasi model jaringan.

4.1 Data Collection

Tahap 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 peniltian 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 pemograman 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
(Sumber: Dokumen Penulis)

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*

(Sumber: Dokumen Penulis)

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 [11].

4.2.2 **Menghapus *Tweet* yang Duplikat**

Data *tweet* yang telah dikumpulkan pada tahapan *scraping* memungkinkan masih terdapat data yang identik atau 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
(Sumber: Dokumen Penulis)

Kode Program 4.3 adalah *script* Python untuk menghapus salah satu dari data yang identik atau 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 untuk 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 (Sumber: Dokumen Penulis)

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 *node* yang *self-loop*. Sehingga, penghapusan *tweet* dengan kriteria ini akan memudahkan tahapan analisis selanjutnya.

4.2.4 Mengambil *Tweet* Percakapan Antara Konsumen

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

```
# 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 Konsumen (Sumber: Dokumen Penulis)

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 konsumen.

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 Perusahaan *Edtech* (Sumber: Dokumen Penulis)

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 *nodes* 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())
    # weight column
    edglst = edglst.value_counts(
        ).to_frame('Weight').astype(float)
    return edglst.reset_index()
# implement 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*

(Sumber: Dokumen Penulis)

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 *nodes* 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 membuat atribut *Weight* atau bobot pada *edge list* berdasarkan jumlah interaksi yang identik antara sepasang *nodes*. 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* untuk Ruangguru dan Zenius menggunakan pendekatan *Social Network Analysis* (SNA). Pada tahapan ini, penulis menggunakan *library Networkx* dari bahasa pemrograman Python.

```
g_ruangguru , g_zenius = [
    nx.from_pandas_edgelist(
        x,
        source='Source',
        target='Target', edge_attr='Weight')
    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 *Undirected Graph*
(Sumber: Dokumen Penulis)

Kode Program 4.8 adalah *script* Python untuk melakukan transformasi *edge list* ke bentuk *undirected graph*. *Undirected graph* dipilih karena pada penelitian ini penulis hanya berfokus untuk menganalisis persebaran informasi berdasarkan adanya interaksi antara suatu *node* dengan *node* lainnya. Sehingga, arah interaksi atau urutan *nodes* 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* [12].

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* atau banyaknya *nodes* dan nilai *size* atau banyaknya *edges* 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 dari Graf*
(Sumber: Dokumen Penulis)

Kode Program 4.9 adalah *script* Python untuk menghitung nilai *order* dan *size* pada masing-masing graf. Nilai *order* dan *size* pada graf Ruangguru masing-masing adalah 4.982 dan 5.488, sedangkan untuk graf 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* dari Graf

(Sumber: Dokumen Penulis)

Kode Program 4.10 adalah *script* Python untuk menghitung nilai *density* pada masing-masing graf. Nilai *density* untuk graf Ruangguru adalah 0,00033089 sedangkan nilai *density* untuk graf 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 [13].

```
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 Graf
(Sumber: Dokumen Penulis)

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

4.3.1.4 Diameter

Untuk menghitung nilai *diameter* atau *shortest path* terjauh antara sepasang *nodes* 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 Graf
(Sumber: Dokumen Penulis)

Kode Program 4.12 adalah *script* Python untuk menghitung nilai *diameter* pada graf. Pertama, penulis mencari jarak *shortest path* pada seluruh pasang *nodes* 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 graf Ruangguru adalah 19 dan nilai *diameter* untuk graf 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 *nodes*.

4.3.1.5 Average Path Length

Untuk menghitung nilai *average path length* atau rata-rata *shortest path* pada setiap pasang *nodes* 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 Graf

(Sumber: Dokumen Penulis)

Kode Program 4.13 adalah *script* Python untuk menghitung nilai *average path length* pada graf. Pertama, penulis mencari jarak *shortest path* pada seluruh pasang *nodes* di jaringan menggunakan algoritma BFS. Selanjutnya, dari seluruh jarak *shortest path* tersebut, dicari nilai rata-ratanya dengan menggunakan fungsi *sum* untuk mencari nilai totalnya lalu dibagi dengan banyaknya *nodes* pada jaringan dengan menggunakan fungsi *order*. Nilai *average path length* untuk graf Ruangguru adalah 1,7017 dan nilai *average path length* untuk graf 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 *edges* yang menghubungkan suatu *node* ke *node* 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 Graf
(Sumber: Dokumen Penulis)

Kode Program 4.14 adalah *script* Python untuk menghitung nilai *average degree* pada graf. Pertama, penulis mencari nilai *degree* pada setiap *nodes* di jaringan. Selanjutnya, dari nilai tersebut, dicari nilai rata-ratanya dengan menggunakan fungsi *sum* untuk mencari nilai totalnya lalu dibagi dengan banyaknya *nodes* pada jaringan dengan menggunakan fungsi *order*. Nilai *average degree* untuk graf Ruangguru adalah 1,8156 dan nilai *average degree* untuk graf 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 daripada jaringan percakapan mengenai Zenius karena satu *node* dapat menyebarkan informasi menuju lebih banyak *nodes* lainnya.

4.3.1.7 *Connected Components*

Untuk menghitung nilai *connected components* atau kelompok *nodes* 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 Graf
(Sumber: Dokumen Penulis)

Kode Program 4.15 adalah *script* Python untuk menghitung nilai *connected components* pada graf. Nilai *connected components* dari graf Ruangguru adalah 1.022 dan nilai *connected components* dari graf 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, *nodes* 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 *Network Properties* pada Graf 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

(Sumber: Dokumen Penulis)

Tabel 4.1 menunjukkan perbandingan seluruh hasil perhitungan metrik *network properties* pada graf Ruangguru dan Zenius. Terlihat bahwa karakteristik jaringan percakapan pengguna Twitter mengenai Ruangguru unggul dalam tiga kategori pada metrik *network properties* yaitu *size*, *order*, dan *average degree*. Sedangkan, karakteristik jaringan percakapan pengguna Twitter mengenai Zenius unggul dalam lima kategori pada metrik *network properties* yaitu *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 *node* yang merupakan pusat penyebaran informasi (*key actor*) di antara semua *nodes* yang ada 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 *edges* yang dimiliki *nodes* 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_degreetcent_ruangguru, df_degreetcent_zenius
= [degree_c(x) for x in
[g_ruangguru,g_zenius]]
```

Kode Program 4.16 Degree Centrality pada Graf
(Sumber: Dokumen Penulis)

Kode Program 4.16 adalah *script* Python untuk menghitung nilai *degree centrality* pada seluruh *nodes* di masing-masing jaringan. *Output* dari kode di atas adalah berupa dua *dataframes* yang berisi kumpulan *nodes* beserta nilai *degree centrality* yang dimiliki kumpulan *nodes* 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 *Degree Centrality* Tertinggi pada Graf Ruangguru

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

(Sumber: Dokumen Penulis)

Tabel 4.2 menunjukkan lima *nodes* teratas pada metrik *degree centrality* untuk graf Ruangguru. Kelima *nodes* 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 *Degree Centrality* Tertinggi pada Graf Zenius

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

(Sumber: Dokumen Penulis)

Tabel 4.3 menunjukkan lima *nodes* teratas pada metrik *degree centrality* untuk graf Zenius. Kelima *nodes* tersebut adalah akun schfess, sbmptnfess, subschfess, sabdaps, dan zenius_oliv. Kelima akun tertinggi pada metrik *degree centrality* tersebut merupakan akun yang mempunyai *social connection* tertinggi pada masing-masing graf.

4.3.2.2 *Betweenness Centrality*

Untuk menghitung nilai *betweenness centrality* atau proporsi *shortest path* di antara semua pasangan *nodes* di dalam jaringan yang melewati suatu *node* tertentu, penulis menggunakan fungsi *betweenness centrality* dari library Networkx.

```
# betweenness centrality function
def betweenness_c(g):
    res =
nx.algorithms centrality.betweenness_centrali
ty(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 Graf

(Sumber: Dokumen Penulis)

Kode Program 4.17 adalah *script* Python untuk menghitung nilai *betweenness centrality* pada seluruh *nodes* di masing-masing jaringan. *Output* dari kode di atas adalah berupa dua *dataframes* yang berisi kumpulan *nodes* beserta nilai *betweenness centrality* yang dimiliki kumpulan *nodes* 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 *Betweenness Centrality* Tertinggi pada Graf Ruangguru

No.	Node	<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

(Sumber: Dokumen Penulis)

Tabel 4.4 menunjukkan lima *nodes* teratas pada metrik *betweenness centrality* untuk graf Ruangguru. Kelima *nodes* 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 *Betweenness Centrality* Tertinggi pada Graf Zenius

No.	Node	<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

(Sumber: Dokumen Penulis)

Tabel 4.5 menunjukkan lima *nodes* teratas pada metrik *betweenness centrality* untuk graf Zenius. Kelima *nodes* tersebut adalah akun schfess, sbmptnfess, subschfess, zenius_oliv, dan hopefullyperf. Kelima akun tersebut merupakan elemen sentral dalam lalu lintas informasi di masing-masing jaringan.

4.3.2.3 Closeness Centrality

Untuk menghitung nilai *closeness centrality* atau rata-rata pada semua *shortest path* dari suatu *node* ke setiap *nodes* lain 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 Graf
(Sumber: Dokumen Penulis)

Kode 4.18 Program adalah *script* Python untuk menghitung nilai *closeness centrality* pada seluruh *nodes* di masing-masing jaringan. *Output* dari kode di atas adalah berupa dua *dataframes* yang berisi kumpulan *nodes* beserta nilai *closeness centrality* yang dimiliki kumpulan *nodes* 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 *Closeness Centrality* Tertinggi pada Graf Ruangguru

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

(Sumber: Dokumen Penulis)

Tabel 4.6 menunjukkan lima *nodes* teratas pada metrik *closeness centrality* untuk graf Ruangguru. Kelima *nodes* 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 *Closeness Centrality* Tertinggi pada Graf Zenius

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

(Sumber: Dokumen Penulis)

Tabel 4.7 menunjukkan lima *nodes* teratas pada metrik *closeness centrality* untuk graf Zenius. Kelima *nodes* tersebut adalah akun schfess, zenius_oliv, hopefullyperf, byunpov, dan amsterdamlaf. Kelima akun tersebut merupakan *node* yang paling dekat dengan semua *nodes* lainnya di masing-masing jaringan.

4.3.2.4 Eigenvector Centrality

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

(Sumber: Dokumen Penulis)

Kode Program 4.19 adalah *script* Python untuk menghitung nilai *eigenvector centrality* pada seluruh *nodes* di masing-masing jaringan. *Output* dari kode di atas adalah berupa dua *dataframes* yang berisi kumpulan *nodes* beserta nilai *eigenvector centrality* yang dimiliki kumpulan *nodes* 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 *Eigenvector Centrality* Tertinggi pada Graf Ruangguru

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

(Sumber: Dokumen Penulis)

Tabel 4.8 menunjukkan lima *nodes* teratas pada metrik *eigenvector centrality* untuk graf Ruangguru. Kelima *nodes* 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 *Eigenvector Centrality* Tertinggi pada Graf Zenius

No.	Node	<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

(Sumber: Dokumen Penulis)

Tabel 4.9 menunjukkan lima *nodes* teratas pada metrik *eigenvector centrality* untuk graf Zenius. Kelima *nodes* tersebut adalah akun schfess, subschfess, sbmptnfess, zenius_oliv, dan hopefullyperf. Kelima akun tersebut merupakan *node* yang mempunyai kuantitas dan kualitas koneksi terbaik di masing-masing jaringan.

Tabel 4.10 Seluruh Metrik *Centrality* pada Graf Ruangguru

<i>Node</i>	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

(Sumber: Dokumen Penulis)

Tabel 4.10 menunjukkan seluruh nilai metrik *centrality* pada graf Ruangguru. Dari keempat metrik tersebut, terlihat bahwa *nodes* dengan nama akun schfess, subschfess, dan sbmptnfess merupakan *nodes* yang selalu menempati lima besar teratas. Hal ini menunjukkan bahwa ketiga akun tersebut merupakan *key actor* dalam penyebaran informasi pada graf Ruangguru.

Tabel 4.11 Seluruh Metrik *Centrality* pada Graf Zenius

<i>Node</i>	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

(Sumber: Dokumen Penulis)

Tabel 4.11 menunjukkan seluruh nilai metrik *centrality* pada graf Zenius. Dari keempat metrik tersebut, terlihat bahwa *nodes* dengan nama akun schfess dan zenius_oliv merupakan *nodes* yang selalu menempati lima besar teratas. Hal ini menunjukkan bahwa kedua akun tersebut merupakan *key actor* dalam penyebaran informasi pada graf 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 wawasan tentang hasil analisis jaringan. Data *edge list* beserta hasil perhitungan metrik-metrik yang telah dilakukan pada tahapan sebelumnya, akan diolah menggunakan perangkat lunak visualisasi graf yaitu Gephi. Namun, sebelum itu, penulis akan melakukan tranformasi data graf Networkx ke bentuk *file Graph Exchange XML Format (GEXF)* [14]

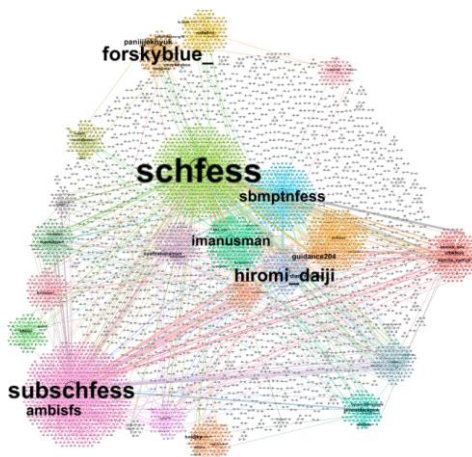
```
# adding attributes to graph function
def data_to_gephi(g):
    nx.set_node_attributes(
        g,
        community.best_partition(
            g, resolution=1),
        "Modularity Class")
    nx.set_node_attributes(
        g,
        nx.betweenness_centrality(
            g),
        "Betweenness 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* (Sumber: Dokumen Penulis)

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 [14].

Terdapat dua atribut pada graf 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* atau *node* yang berperan sentral dalam penyebaran informasi di masing-masing kelompok dan di dalam jaringan.

Selanjutnya, kedua *files* GEXF tersebut diolah dalam aplikasi Gephi. Jenis *layout* visualisasi yang digunakan penulis adalah *circle pack layout*.



Gambar 4.1 Visualisasi Graf Ruangguru
(Sumber: Dokumen Penulis)

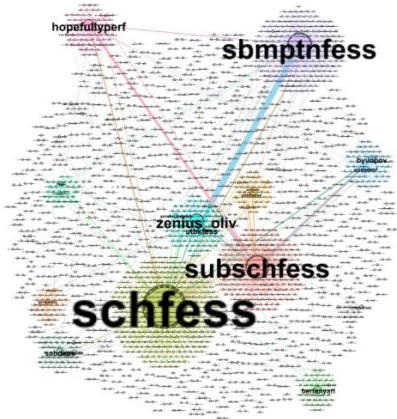
Gambar 4.1 adalah visualisasi graf Ruangguru. Terdapat 20 kelompok besar di dalam jaringan (kelompok yang diberikan warna). Kelompok yang besar adalah kelompok yang mempunyai persentase anggota di atas atau sama dengan 1,00 %. Daftar lima kelompok terbesar pada graf Ruangguru beserta atributnya ditunjukkan pada Tabel 4.12.

Tabel 4.12 Lima Kelompok Terbesar pada Graf Ruangguru

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

(Sumber: Dokumen Penulis)

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



Gambar 4.2 Visualisasi Graf Zenius

(Sumber: Dokumen Penulis)

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). Kelompok yang besar adalah kelompok yang mempunyai persentase anggota di atas atau sama dengan 1,00 %. 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 Graf 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

(Sumber: Dokumen Penulis)

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 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 *nodes* di dalam kelompok masing-masing, *key actor* tersebut juga berpengaruh dalam penyebaran informasi pada seluruh *nodes* di dalam jaringan. Oleh karena itu, fakta ini semakin menguatkan bahwa *key actor* pada graf Ruangguru adalah akun schfess, subschfess, dan sbmptnfess. Sedangkan, *key actor* untuk graf Zenius adalah akun schfess dan zenius_oliv.

“Halaman ini sengaja dikosongkan”

BAB V

KESIMPULAN DAN SARAN

Pada bab ini berisi kesimpulan dari penelitian yang telah dilakukan serta saran dari penulis agar penelitian dapat dikembangkan lebih baik.

5.1. Kesimpulan

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

1. Implementasi SNA dalam menganalisis karakteristik suatu jaringan dapat menggunakan metrik *network properties*. Berdasarkan hasil perhitungan dan perbandingan metrik *network properties*, jaringan percakapan pengguna Twitter mengenai perusahaan *edtech* yang paling banyak unggul adalah Zenius dengan total lima dari delapan metrik. Graf 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 menganalisis pengguna Twitter yang paling berpengaruh (*key actor*) dalam penyebaran informasi pada suatu jaringan dapat menggunakan metrik *centrality*. Melalui perhitungan metrik *degree centrality*, *betweenness centrality*, *closeness centrality*, dan *eigenvector centrality* didapatkan hasil *key actor* pada masing-masing jaringan. *Key actor* pada graf Ruangguru adalah akun *schfess*, *subschfess*, dan *sbmptnfess*.

Sedangkan, *key actor* untuk graf Zenius adalah akun schfess dan zenius_oliv. Selain di dalam jaringan, akun-akun tersebut 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 *nodes* di dalam kelompok, *key actor* tersebut juga berpengaruh dalam penyebaran informasi pada seluruh *nodes* di dalam jaringan.

5.2. Saran

Berdasarkan hasil analisis dan penelitian pada bab sebelumnya, terdapat beberapa saran bagi perusahaan *edtech*:

1. Akun Twitter perusahaan dapat lebih aktif lagi melakukan *engagement* atau interaksi dengan *followers*. 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. Akun-akun *key actors* pada jaringan Ruangguru dan Zenius merupakan 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 *node* yang mempunyai nilai metrik *centrality* yang cukup tinggi pada masing-masing jaringan adalah *node* 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

- [1] B. Riaz, N. Yarrow, and M. Cali, “EdTech in Indonesia: Ready for Take-off,” *World Bank, Washington DC*, 2020.
- [2] I. Antoniadis and A. Charmantzi, “Social network analysis and social capital in marketing: theory and practical implementation,” *International Journal of Technology Marketing*, vol. 11, p. 344, Jan. 2016, doi: 10.1504/IJTMKT.2016.077387.
- [3] A. Litterio, E. A. Nantes, J. Larrosa, and L. Gómez, “Marketing and social networks: a criterion for detecting opinion leaders,” *European Journal of Management and Business Economics*, vol. 26, pp. 347–366, Oct. 2017, doi: 10.1108/EJMBE-10-2017-020.
- [4] I. Himelboim and G. Golan, “A Social Networks Approach to Viral Advertising: The Role of Primary, Contextual, and Low Influencers,” *Social Media + Society*, vol. 5, p. 205630511984751, Jul. 2019, doi: 10.1177/2056305119847516.
- [5] “Aplikasi Bimbingan Belajar Online #1 di Indonesia | Ruangguru.” <https://www.ruangguru.com/> (accessed Sep. 23, 2021).
- [6] “About - Zenius Education.” <https://www.zenius.net/about/> (accessed Sep. 23, 2021).
- [7] A. Mollett, D. Moran, and P. Dunleavy, “Using Twitter in university research, teaching and impact activities,” 2011.
- [8] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.
- [9] “Network Science by Albert-László Barabási.” <http://networksciencebook.com/> (accessed Nov. 09, 2021).

- [10] A. Fornito, A. Zalesky, and E. Bullmore, *Fundamentals of brain network analysis*. Academic Press, 2016.
- [11] “pandas documentation — pandas 1.3.5 documentation.” <https://pandas.pydata.org/docs/> (accessed Jan. 18, 2022).
- [12] “Software for Complex Networks — NetworkX 2.6.2 documentation.” <https://networkx.org/documentation/stable/index.html> (accessed Jan. 18, 2022).
- [13] “Community detection for NetworkX’s documentation — Community detection for NetworkX 2 documentation.” <https://python-louvain.readthedocs.io/en/latest/> (accessed Jan. 18, 2022).
- [14] “Learn how to use Gephi.” <https://gephi.org/users/> (accessed Jan. 19, 2022).

“Halaman ini sengaja dikosongkan”

LAMPIRAN

Lampiran 1

Tabel Deskripsi Atribut pada Data Mentah

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>	<i>Weight</i>
7chillboo	kjnchsolo	1440
morphoflies	junkyukime	1040
exosalien	junkyukime	1000
mactaadid	kyutieshii	550
yourarians	sbmptnfess	360
mumarisatulhk	schfess	360
dian_renataa	schfess	350
helloyoon4	sbmptnfess	340
guidance204	schfess	340
syafirainurainun	schfess	270
guidance204	subschfess	250
mumarisatulhk	subschfess	220
jeinneblackpink	subschfess	210
dian_renataa	subschfess	210
⋮	⋮	⋮
hasahihasaki	hyunsuksis_	10
hasahihasaki	bxxxyedam	10
hasahihasaki	ailurocfie	10
harvkyvsvk	swidays7	10
harvezmoon	taytawanreal	10
haruvairy	yoshaurs	10
harutodongaeng	misellia_	10
harutodongaeng	justjustinpark	10
harutobaby__	reinc_	10
harutoair	ambisfs	10
zzzcapricorn	convomf	10

Lampiran 3

Tabel *Edge List* Zenius

<i>Source</i>	<i>Target</i>	<i>Weight</i>
helloyon4	sbmptnfess	340
helloyon4	utbkfess	210
byunpov	subschfess	150
hopefullyperf	subschfess	140
byunpov	schfess	110
lizypeachyy	schfess	100
zenius_oliv	subschfess	90
gabriux1	dinges_zenius	90
zeniusambis	schfess	80
hopefullyperf	schfess	80
helloyon4	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	prkdllx	10
hiraahero	itzjaraaa	10

Lampiran 4

Tabel Metrik *Centrality* pada Graf Ruangguru

No.	Node	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	beeerdebuuu	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	ydefay	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 Graf Zenius

No.	Node	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 Graf Ruangguru

<i>Node</i>	<i>Modularity Class</i>
7chillboo	0
kjnchsolo	0
morphoflies	1
junkyukime	1
exosalien	1
mactaadid	2
kyutieshii	2
yourarians	3
sbmptnfess	4
mumarisatulhk	5
schfess	6
dian_renataa	3
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 Graf Zenius

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

BIODATA PENULIS



Adrian Maulana Muhammad, atau biasa dipanggil Adrian, lahir di Ujung Pandang pada tanggal 18 Juli 1996. Pendidikan formal yang sudah ditempuh oleh penulis dimulai dari TK Antam Pomalaa, SDI Al-Ikhlas Jakarta, SMPN 85 Jakarta, dan SMAN 34 Jakarta. Untuk saat ini penulis sedang menempuh Pendidikan S1 di Departemen Matematika, Fakultas Sains dan Analitika Data, Institut Teknologi Sepuluh Nopember. Selama berkuliah penulis mengambil bidang minat Ilmu Komputer, khususnya bidang Pembelajaran Mesin dan *Big Data*. Selama kuliah, penulis aktif mengikuti kepanitiaan *event*, diantaranya OMITS (Olimpiade Matematika ITS) sebagai Ketua Koordinator Akomodasi dan Transportasi (2017). Penulis juga mengikuti beberapa pelatihan pada bidang ilmu komputer, salah satunya adalah Google Developers Kejar (2019). Selain itu penulis juga aktif dalam berbagai pelatihan kemahasiswaan seperti LKMM Pra-TD, LKMM TD, pelatihan PKM GT, Mathematics Heroes School, dan lain sebagainya. 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 adriantoto7@gmail.com.Terimakasih.