

# Analisis Interaksi Pengguna Twitter Mengenai Bisnis *Educational Technology* Menggunakan Pendekatan *Social Network Analysis* (Studi Kasus: Ruangguru Dan Zenius)

Adrian Maulana Muhammad, Dr. Imam Mukhlash, S.Si, MT, dan Dr. Darmaji, S.Si, MT  
Departemen Matematika, Fakultas Sains dan Analitik Data,  
Institut Teknologi Sepuluh Nopember (ITS)  
Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia  
e-mail: adriantoto7@gmail.com

**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.

## I. PENDAHULUAN

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. Berdasarkan data *World Bank*, peningkatan pada pendirian perusahaan *edtech* juga bertepatan dengan peningkatan penetrasi internet di Indonesia [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]. Ruangguru dan Zenius juga merupakan akun platform *edtech* di Indonesia yang mempunyai pengikut terbanyak di situs jejaring sosial *online* Twitter, dan perusahaan hal tersebut bisa menjadi peluang untuk perusahaan dalam menjalin hubungan dengan konsumen serta melakukan 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).

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].

Berdasarkan latar belakang yang telah diuraikan di atas, 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*.

## II. METODOLOGI PENELITIAN

### A. Data Penelitian

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 30.867 *tweets* dengan

8.152 *nodes* dan 26.803 *edges* untuk Ruangguru dan 3.118 *nodes* dan 3.026 *edges* untuk Zenius.

## B. Data Collection

Pengumpulan data dilakukan dengan cara *scraping* dari media sosial Twitter. Tahapan ini dilakukan dengan menggunakan bahasa pemrograman Python dan *library* Twint. Data yang akan dilakukan *scraping* merupakan data yang dapat diakses bebas oleh publik.

## C. Data Pre-processing

Seluruh *tweet* yang sudah terkumpul, kemudian dilakukan tahap *data cleaning* untuk menghilangkan *tweet* yang tidak relevan agar lebih mudah untuk dilakukan proses analisis. Selanjutnya, dilakukan tahap *data transformation* ke dalam bentuk *edge list* dengan bantuan *package* Pandas dan Networkx dari bahasa pemrograman Python.

## D. Analisis Social Network Analysis

*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 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* [5].

### 1. Network Properties

Setiap model jaringan yang sudah diproses dengan aplikasi Gephi, memiliki beberapa *properties* yang akan dihitung nilainya, sebagai berikut:

#### 1) Order dan Size

Pada suatu jaringan, *order* adalah banyaknya *nodes* dan *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* [5].

#### 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 [5]:

$$\text{Network Density} = \frac{\text{Total Edges}}{\text{Total Possible Edges}} = \frac{m}{n(n-1)/2} \quad (2.1)$$

Keterangan:

- m* : jumlah *edges* pada jaringan
- n* : jumlah *nodes* pada jaringan

### 3) Modularity

*Modularity* merupakan metrik yang digunakan untuk mengetahui kualitas pembagian jaringan ke dalam kelompok/klaster. 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*. Perubahan nilai modularitas ( $\Delta M$ ) dapat dihitung dengan formula sebagai berikut [5]:

$$\Delta M = \left[ \frac{\sum_{in} + 2k_{i,in}}{2m} - \left( \frac{\sum_{tot} + k_i}{2m} \right)^2 \right] - \left[ \frac{\sum_{in}}{2m} - \left( \frac{\sum_{tot}}{2m} \right)^2 - \left( \frac{k_i}{2m} \right)^2 \right] \quad (2.2)$$

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

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 [5].

### 4) Diameter

*Diameter* adalah *shortest path* terpanjang 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*. Untuk jaringan yang 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 [5]:

- Mulai dari *node i* yang kita beri label “0”.
- Temukan *node* yang terhubung langsung ke *i*. Beri label “1” dan tempatkan di dalam antrian.
- 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.
- Ulangi langkah 3 sampai anda menemukan *node j* yang merupakan target *node* atau tidak ada lagi *node* dalam antrian.
- Jarak antar *i* dan *j* adalah label untuk *j*. Jika *j* tidak mempunyai label, maka  $d_{ij} = \infty$ .

### 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. Formula untuk menghitung *average path length* adalah sebagai berikut [5].

$$\langle d \rangle = \frac{2}{n(n-1)} \sum_{\substack{i,j=1,n \\ i \neq j}} d_{i,j} \quad (2.3)$$

Keterangan:

- n* : jumlah *nodes* pada jaringan  
 $d_{i,j}$  : bobot *shortest path* antara *node i* dan *node j*

### 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 [5]:

$$\langle k \rangle = \frac{1}{n} \sum_{i=1}^n k_i \quad (2.4)$$

Keterangan:

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

### 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.

## 2. Centrality

Setelah metrik *network properties*, metrik selanjutnya adalah *centrality*. Pengukuran pada *centrality* bertujuan untuk mengidentifikasi *node* yang merupakan pusat komunikasi di

antara semua *nodes* yang ada dalam suatu jaringan. Terdapat empat pengukuran *centrality* pada penelitian ini, yaitu:

#### 1) Degree Centrality

*Degree centrality* menggambarkan ukuran *social connections* yang dipunyai *node* dalam jaringan. *Node* dengan nilai *degree centrality* tertinggi bisa saja merupakan inti dari jaringan, namun bisa saja berada jauh di tepi jaringan. Berikut adalah formula untuk *degree centrality* untuk *node i* [6]:

$$C_D(i) = \sum_{j \neq i}^n a_{ij} \quad (2.5)$$

$$a_{ij} = \begin{cases} 1, & \text{apabila terdapat edge di antara node } i \text{ dan } j \\ 0, & \text{yang lainnya} \end{cases} \quad (2.6)$$

Keterangan:

- n* : jumlah *nodes* pada jaringan

#### 2) Betweenness Centrality

*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. Untuk menghitung *betweenness centrality* pada suatu *node i*, kita menghitung proporsi *shortest paths* antara *node j* dan *h* yang melewati *i* [6].

$$C_B(i) = \frac{1}{(n-1)(n-2)} \cdot \sum_{h \neq i, h \neq j, j \neq i}^n \frac{\rho_{hj}(i)}{\rho_{hj}} \quad (2.7)$$

Keterangan:

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

#### 3) Closeness Centrality

*Closeness centrality* adalah mencari *node* yang paling dekat dengan semua *nodes* lainnya. *Closeness centrality* untuk suatu *node* adalah rata-rata pada semua *shortest paths* dari *node* tersebut ke setiap *nodes* lain dalam jaringan. Formula untuk menghitung *closeness centrality* pada *node i* adalah sebagai berikut [6]:

$$C_c(i) = \frac{n-1}{\sum_{j \neq i} l_{ij}} \quad (2.8)$$

Keterangan:

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

#### 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.

Untuk memperkirakan ukuran ini, kita harus mempertimbangkan *eigenvalues* dan *eigenvectors* dari matriks ketetanggaan. Untuk mencari *eigenvalues* maka menggunakan persamaan karakteristik polinomial sebagai berikut [6]:

$$|A - \lambda I| = 0 \quad (2.9)$$

Keterangan:

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

Selanjutnya yaitu mencari *eigenvector* dengan menggunakan *eigenvalue* terbesar, sebagai berikut [6]:

$$A\vec{v} = \lambda\vec{v} \quad (2.10)$$

$$(A - \lambda I)\vec{v} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \quad (2.11)$$

Keterangan:

- $A$  : matriks ketetanggaan berukuran  $n \times n$   
 $\lambda$  : *eigenvalue* terbesar  
 $\vec{v}$  : *eigenvector*  
 $I$  : matriks identitas

Notasi  $\vec{v}$  merupakan *eigenvector* matriks  $n \times 1$  yang dapat direpresentasikan sebagai persamaan berikut [6]:

$$\vec{v} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad (2.15)$$

*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 nilainya yang tertinggi.

### E. Visualiasi Model Jaringan

Data edge list kemudian diproses kembali menggunakan aplikasi Gephi untuk dibuat visualisasi model jaringannya, dengan menggunakan graf dengan jenis *undirected graph*, yaitu graf yang tidak memperhitungkan orientasi arah hubungan antar *nodes* yang dimiliki jaringan.

## III. ANALISIS DAN PEMBAHASAN

### A. Data Collection

Pengambilan data dilakukan dengan metode *scraping*. *Scraping* merupakan teknik pengambilan atau ekstraksi data dari suatu *website*, lalu data tersebut disimpan dalam format *file Comma-Separated Values* (CSV). Pada penelitian ini, proses

*web scraping* dilakukan dengan menggunakan *library* Twint dari bahasa pemrograman Python.

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* CSV yang berisi data *tweet* hasil *scraping*. Ukuran data mentah yang digunakan pada penelitian ini adalah 39.219 baris dan 36 kolom.

### B. 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.

#### 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.

#### 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.

#### 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*. 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) 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. Tahapan ini dilakukan karena *tweet* yang akan dianalisis adalah interaksi percakapan hanya antara konsumen.

#### 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. 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.

#### 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. 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.

### C. *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 untuk melakukan transformasi *edge list* ke bentuk *undirected graph* dan perhitungan nilai metrik. *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.

#### 1) Perhitungan Metrik *Network Properties*

- *Size dan Order*

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.

- *Density*

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.

- *Modularity*

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.

- *Diameter*

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*.

- *Average Path Length*

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.

- *Average Degree*

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.

- *Connected Components*

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 1.

**Tabel 1** *Network Properties* pada Graf Ruangguru dan Zenius

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

(Sumber: Dokumen Penulis)

Tabel 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*.

## 2) Perhitungan Metrik Centrality

### • Degree Centrality

Tabel 2 menunjukkan lima *nodes* teratas pada metrik *degree centrality* untuk graf Ruangguru. Kelima *nodes* tersebut adalah akun schfess, subschfess, ambisfs, sbmptnfess, dan guidance204.

**Tabel 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 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.

**Tabel 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)

### • Betweenness Centrality

Tabel 4 menunjukkan lima *nodes* teratas pada metrik *betweenness centrality* untuk graf Ruangguru. Kelima *nodes* tersebut adalah akun schfess, subschfess, forskyblue, hiromi\_daiji, dan sbmptnfess.

**Tabel 4** *Betweenness Centrality* Teratas pada Graf Ruangguru

No.	Node	Betweenness Centrality
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 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.

**Tabel 5** *Betweenness Centrality* Tertinggi pada Graf Zenius

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

(Sumber: Dokumen Penulis)

### • Closeness Centrality

Tabel 6 menunjukkan lima *nodes* teratas pada metrik *closeness centrality* untuk graf Ruangguru. Kelima *nodes* tersebut adalah akun schfess, hiromi\_daiji, subschfess, syafiranurainun, dan sbmptnfess.

**Tabel 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 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.

**Tabel 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)

- *Eigenvector Centrality*

Tabel 8 menunjukkan lima *nodes* teratas pada metrik *eigenvector centrality* untuk graf Ruangguru. Kelima *nodes* tersebut adalah akun schfess, subschfess, ambisfs, sbmptnfess, dan guidance204.

**Tabel 8** *Eigenvector Centrality* Teratas pada Graf Ruangguru

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

(Sumber: Dokumen Penulis)

Tabel 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 9** *Eigenvector Centrality* Tertinggi pada Graf Zenius

No.	Node	Eigenvector Centrality
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 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.

**Tabel 10** Seluruh Metrik *Centrality* pada Graf Ruangguru

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

(Sumber: Dokumen Penulis)

Tabel 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 akun-akun tersebut merupakan *key actor* dalam penyebaran informasi pada masing-masing graf.

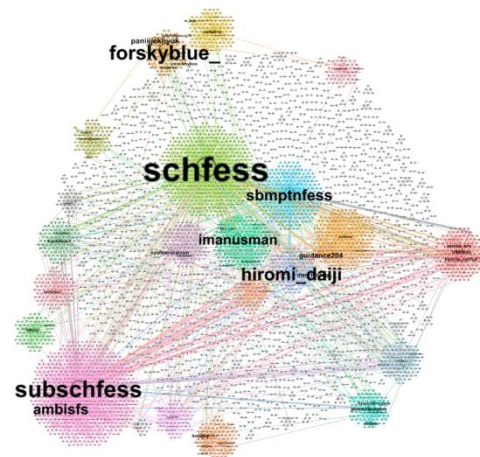
**Tabel 11** Seluruh Metrik *Centrality* pada Graf Zenius

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

(Sumber: Dokumen Penulis)

#### D. 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. Terdapat dua atribut pada graf yang akan membantu dalam pengolahan visualisasi, yaitu atribut pada metrik *modularity* dan *betweenness centrality*. Metrik *modularity* dibutuhkan pada tahap visualisasi karena dapat menunjukkan kelompok atau kluster yang terbentuk pada jaringan dan metrik *betweenness centrality* dibutuhkan pada tahap visualisasi karena dapat menunjukkan *key actor* atau *node* yang berperan sentral dalam penyebaran informasi di masing-masing kelompok dan di dalam jaringan.

**Gambar 1** Visualisasi Graf Ruangguru

(Sumber: Dokumen Penulis)

Gambar 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

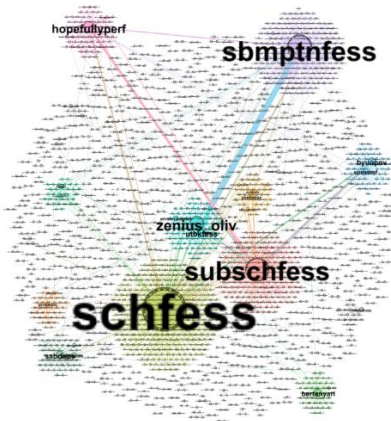


persen. Daftar lima kelompok terbesar pada graf Ruangguru beserta atributnya ditunjukkan pada Tabel 12.

**Tabel 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)



**Gambar 2** Visualisasi Graf Zenius

(Sumber: Dokumen Penulis)

Gambar 2 adalah visualisasi model jaringan percakapan pengguna Twitter mengenai Zenius. Terdapat 11 kelompok atau kluster besar di dalam jaringan. Daftar lima kelompok terbesar pada jaringan percakapan pengguna Twitter mengenai Zenius beserta atributnya ditunjukkan pada Tabel 13.

**Tabel 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 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.

## IV. KESIMPULAN DAN SARAN

### A. Kesimpulan

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

1. Berdasarkan hasil perhitungan dan perbandingan metrik *network properties*, jaringan yang 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. Melalui perhitungan metrik *centrality* didapatkan *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.

### B. 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 *imamusman*. 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. Melakukan proses SNA secara *real-time*, sehingga perusahaan dapat terus memantau kualitas jaringan sosial dengan *followers* yang mereka miliki setiap waktu.
3. 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] "Network Science by Albert-László Barabási." <http://networksciencebook.com/> (accessed Nov. 09, 2021).
- [6] A. Fornito, A. Zalesky, and E. Bullmore, *Fundamentals of brain network analysis*. Academic Press, 2016. B. Smith, "An approach to graphs of linear forms (Unpublished work style)," belum dipublikasikan.
- [7] "Software for Complex Networks — NetworkX 2.6.2 documentation." <https://networkx.org/documentation/stable/index.html> (accessed Jan. 18, 2022).