

Community Detection Menggunakan Genetic Algorithm dalam Social Network Twitter

Community Detection using Genetic Algorithm in Social Network Twitter

Itsna Alfin Nur¹, Moch. Arif Bijaksana², Ir. M.Tech., Eko Darwiyanto, S.T., M.T³.

Prodi S1 Teknik Informatika, Fakultas Informatika, Universitas Telkom

alpin.nur07@gmail.com¹, kaprodiss1if@gmail.com², ekodarwiyanto@telkomuniversity.ac.id³

Abstrak

Twitter telah berkembang dengan pesat menjadi situs *social network* populer dan telah memiliki jutaan pengguna yang tersebar di berbagai belahan dunia. *Twitter* saat ini tidak hanya sekedar digunakan untuk media pertemanan, namun seringkali juga digunakan sebagai media bertukar informasi, media iklan sebuah produk, media kampanye partai-partai politik, dan media propaganda bagi kelompok-kelompok yang memiliki tujuan tertentu. Perkembangan *social network* yang pesat menjadi bahan penelitian yang menarik di kalangan penggiat ilmu informatika, salah satunya adalah *community detection* pada jaringan kompleks seperti *social network*. Dalam *community detection*, tujuannya adalah untuk membagi jaringan ke dalam daerah-daerah pada graph. Daerah tersebut biasanya sesuai dengan entitas-entitas yang memiliki hubungan erat, dan dapat dikatakan sebagai satu komunitas. Salah satu algoritma yang dapat digunakan untuk *community detection* adalah Algoritma yang di dasarkan pada *Genetic Algorithm*. *Genetic Algorithm* dipilih dalam penelitian ini karena dapat menemukan komunitas terbaik dengan memaksimalkan modularitas jaringan. Algoritma ini memiliki $O(e)$ kompleksitas waktu dan tidak perlu mengetahui terlebih dahulu tentang jumlah komunitas atau nilai ambang batas, yang membuat algoritma berguna dalam jaringan di kehidupan nyata. Pembentukan komunitas didasarkan pada nilai *similarity* berdasarkan interaksi antar pengguna *twitter* berupa *mention*, *reply*, *following*, dan *follower*. Hasil akhir dari penelitian ini telah mampu menganalisis pengaruh interaksi berupa *mention*, *reply*, *following*, dan *follower* pada nilai *similarity* dan pengaruh nilai *similarity* tersebut dalam pembentukan komunitas menggunakan *Genetic Algorithm*.

Kata kunci: *Twitter, Social network, Community detection, Genetic Algorithm, Similarity.*

Abstract

Twitter has grown rapidly into a popular *social network* sites and has had millions of users spread across different parts of the world. *Twitter* today is not just used for media friends, but often also used as a medium of exchange of information, media advertising a product, media campaigns of political parties, and media propaganda for groups that have a specific purpose. The rapid development of *social networks* become an interesting research material science informatics among activists, one of whom is a *community detection* in complex networks such as *social network*. In the *community detection*, the goal is to divide the network into areas on the graph. The area usually corresponds to entities that have a close relationship, and can be regarded as one community. One algorithm that can be used for *community detection* algorithm that is based on the *Genetic Algorithm*. *Genetic Algorithm* chosen in this study because it can find the best community in maximizing the network modularity. This algorithm has $O(e)$ the complexity of time and do not need to know in advance about the amount of community or threshold values, which makes the algorithm useful in a network in real life. The establishment of community based on the similarity value is based on the interaction among *twitter* users in the form of *mention*, *reply*, *following* and *follower*. The end result of this study is able to analyze the effect of the interaction in the form of *mention*, *reply*, *following* and *followers* on the value of similarity and the similarity values influence the formation of communities using a *Genetic Algorithm*.

Keywords: *Twitter, Social network, Community detection, Genetic Algorithm, Similarity.*

1. Pendahuluan

1.1 Latar Belakang

Twitter telah berkembang dengan pesat menjadi situs *social network* populer dan telah memiliki jutaan pengguna yang tersebar di berbagai belahan dunia. Sekarang ini pengguna *twitter* telah mencapai 271 juta pengguna aktif perbulan dan perharinya ada 500 juta *tweets* yang dikirim. *Twitter* saat ini tidak hanya sekedar digunakan untuk media pertemanan, namun seringkali juga digunakan sebagai media bertukar informasi, media iklan sebuah produk, media kampanye partai-partai politik, dan media propaganda bagi kelompok-kelompok yang memiliki tujuan tertentu.

Perkembangan *social network* yang pesat menjadi bahan penelitian yang menarik di kalangan penggiat ilmu informatika, salah satunya adalah *community detection* pada jaringan kompleks seperti *social network* [12]. Dalam *community detection*, tujuannya adalah untuk membagi jaringan ke dalam daerah-daerah pada graph. Daerah tersebut biasanya sesuai dengan entitas-entitas yang memiliki hubungan erat, dan dapat dikatakan sebagai satu komunitas [10]. Pembentukan komunitas ini kemudian dapat digunakan untuk berbagai keperluan seperti menargetkan skema pemasaran produk, menghitung kepopuleran suatu barang, mendeteksi isu di masyarakat, mendeteksi jaringan sel teroris, dan masih banyak lagi [10]. Interaksi antar user pada *twitter* terbagi menjadi 4 macam yaitu interaksi *mention*, *reply*, *follower* dan *followed*. Interaksi inilah yang akan dijadikan nilai dasar untuk pembentukan komunitas pada *community detection*.

Beberapa algoritma yang sering digunakan untuk penelitian tentang *community detection* diantaranya adalah algoritma *Girvan-Newman*, *K-L algorithm* dan *hierarchical clustering method based on similarity measurement*. Algoritma tersebut telah menghasilkan komunitas yang baik, namun algoritma tersebut belum mampu menghasilkan komunitas dengan penentuan jumlah komunitas yang objektif. Maka dari itu dibutuhkan algoritma yang dapat membentuk komunitas dengan penentuan jumlah komunitas yang objektif dalam jaringan. Salah satu algoritma yang dapat digunakan untuk *community detection* adalah Algoritma yang di dasarkan pada *Genetic Algorithm*. Suatu algoritma mengharuskan pengelompokan komunitas menetapkan batasan sejauh mana suatu kelompok dapat dianggap sebagai komunitas [12]. *Genetic Algorithm* dipilih dalam penelitian ini karena dapat menemukan komunitas terbaik dengan memaksimalkan modularitas jaringan. Algoritma ini memiliki $O(e)$ kompleksitas waktu dan tidak perlu mengetahui terlebih dahulu tentang jumlah komunitas atau nilai ambang batas, yang membuat algoritma berguna dalam jaringan di kehidupan nyata sehingga algoritma ini lebih objektif dalam menentukan banyaknya komunitas dalam suatu jaringan [12].

1.2 Rumusan Masalah

Perumusan masalah dari penelitian ini adalah :

- Bagaimanakah pengaruh interaksi *mention*, *reply* dan *follow* antar user di *social network twitter* dalam perhitungan *similarity*?
- Bagaimanakah pembentukan komunitas menggunakan *Genetic Algorithm*?
- Bagaimanakah nilai *modularity* dari pembentukan komunitas menggunakan algoritma yang didasarkan pada *Genetic Algorithm*?

1.3 Tujuan

Tujuan dari penelitian ini adalah :

- Menganalisis pengaruh interaksi *mention*, *reply* dan *follow* antar user di *social network twitter* dalam perhitungan *similarity*
- Menganalisis proses pembentukan komunitas yang dibentuk menggunakan *Genetic Algorithm*
- Menganalisis nilai *modularity* dari pembentukan komunitas menggunakan algoritma yang didasarkan pada *Genetic Algorithm*

1.4 Batasan Masalah

Batasan masalah pada penelitian ini adalah :

- Pengambilan dataset *twitter* tidak dibatasi dengan topik tertentu
- Perhitungan kedekatan antar user didasarkan pada interaksi *mention*, *reply*, *follower* dan *followed*
- Isi *tweet*, hash tag, dan lokasi tidak diproses dalam penelitian ini
- Pembentukan komunitas tidak mendefinisikan nama komunitas yang terbentuk

2. Tinjauan Pustaka

2.1 Social Network

Sebuah *social network* adalah suatu struktur sosial yang terdiri dari satu set aktor sosial (seperti individu atau organisasi) dan satu set hubungan antara aktor-aktor tersebut [12]. *Social network* pada umumnya didefinisikan sebagai graf yang mewakili hubungan sosial antar manusia atau organisasi. Setiap *node* (juga disebut aktor atau *vertex*) pada graf mewakili satu orang atau sekelompok orang. Sebuah *edge* menghubungkan dua buah *node* (juga disebut tie) mewakili hubungan antara objek yang diwakili oleh dua buah node. Penggunaan graf untuk merepresentasikan *social network* memudahkan untuk memanipulasi, dan menganalisis informasi struktural yang melekat pada hubungan sosial [10].

2.2 Twitter

Twitter sangat cepat berkembang, gratis dan *social network* yang sangat cepat dan telah muncul sebagai sumber utama informasi. *Twitter* adalah situs jejaring sosial *micro-blogging* yang dimulai pada Maret 2006 dan telah memiliki lebih dari 75 juta pengguna terhitung pada Februari 2014 dan berkembang sangat cepat. *Twitter* juga menduduki peringkat nomor 20 dalam segi popularitas di antara semua situs jaringan sosial global dan mendapatkan peringkat sebagai situs *micro-blogging* yang paling populer. *Twitter* memungkinkan penggunaannya untuk mengirim dan berbagi pesan singkat hingga 140 karakter dengan pengguna *twitter* lainnya [10].

2.3 Community Detection

Masalah yang penting dalam jaringan sosial adalah untuk menemukan komunitas pengguna atau *community detection* berdasarkan isi pesan dan hubungan dengan pengguna lain. Sebuah komunitas adalah pola dengan bobot yang besar pada hubungan internal dan rendah pada hubungan eksternal. Hubungan ini dapat diketahui oleh kesamaan konten antar pengguna, persahabatan di antara mereka dan juga kesamaan lainnya pada data pribadi mereka seperti lokasi, jenis kelamin, usia, dll. struktur kedekatan ini kemudian dapat digunakan untuk berbagai keperluan seperti menargetkan skema pemasaran produk, menghitung kepopuleran suatu barang, mendeteksi isu di masyarakat, mendeteksi jaringan sel teroris, dan lain sebagainya. [10].

2.4 Similarity

Similarity adalah tingkat kesamaan antar pengguna yang didasarkan pada interaksi antar pengguna. Komponen yang digunakan untuk menentukan nilai *similarity* adalah interaksi *mention*, *reply*, *following* dan *follower*. Dari masing-masing relasi tersebut akan digabungkan sehingga menghasilkan matriks *similarity* [12].

2.1.1 Similarity Mentions

Tujuan dari *similarity mentions* adalah untuk menghitung kedekatan antar pengguna berdasarkan interaksi mentions yang dilakukan antar pengguna [12].

Kesamaan mention antar pengguna u_i dan u_j dapat dihitung dengan rumus berikut ini :

$$\text{sim}_{\text{Mention}}(i,j) = \frac{C_{\text{Mention}}}{\sqrt{|R_i|} \sqrt{|R_j|}} + \frac{n_{ij} + n_{ji}}{|r_i| + |r_j|} \quad (2.1)$$

Dimana :

$|R_i|$ = jumlah pengguna yang di mention u_i

$|R_j|$ = jumlah pengguna yang di mention u_j

C_{Mention} = jumlah pengguna yang dimention oleh u_i dan u_j

n_{ij} = berapa kali u_i memention u_j

n_{ji} = berapa kali u_j memention u_i

2.1.2 Similarity Follow

Tujuan dari *similarity follow* adalah untuk menghitung kedekatan antar pengguna berdasarkan struktur *follow* [12].

Kesamaan follow antar pengguna u_i dan u_j dapat dihitung dengan rumus berikut ini :

$$\text{Sim}_{\text{Follow}}(i,j) = \frac{C_{\text{Followed}}}{\sqrt{|f_{\text{followed}_i}|} \sqrt{|f_{\text{followed}_j}|}} + \frac{C_{\text{Follower}}}{|f_{\text{follower}_i}| + |f_{\text{follower}_j}|} \quad (2.2)$$

Dimana :

$|f_{\text{followed}_i}|$ = jumlah followed di u_i

$|f_{\text{followed}_j}|$ = jumlah followed di u_j

$|f_{\text{follower}_i}|$ = jumlah follower di u_i

$|f_{\text{follower}_j}|$ = jumlah follower di u_j

C_{followed} = jumlah mutual followed dari u_i dan u_j

C_{follower} = jumlah mutual follower dari u_i dan u_j

2.1.3 Similarity Reply

Tujuan dari *similarity reply* adalah untuk menghitung kedekatan antar pengguna berdasarkan interaksi *reply* yang dilakukan antar pengguna. Respon yang diberikan oleh pengguna dalam menanggapi *mention* yang mereka terima disebut dengan *reply* [12].

Kesamaan reply antar pengguna u_i dan u_j dapat dihitung dengan rumus berikut ini :

$$\text{Sim}_{\text{Reply}}(i,j) = \frac{C_{\text{Reply}}}{\sqrt{|R_i|} \sqrt{|R_j|}} + \frac{n_{ij} + n_{ji}}{|r_i| + |r_j|} \quad (2.3)$$

Dimana :

$|R_i|$ = jumlah pengguna yang di reply u_i

$|R_j|$ = jumlah pengguna yang di reply u_j

C_{Reply} = jumlah pengguna yang di reply oleh u_i dan u_j

n_{ij} = berapa kali u_i mereply u_j

n_{ji} = berapa kali u_j mereply u_i

2.1.4 Similarity Total

Setelah *similarity mentions*, *follow*, dan *reply* di hitung, dalam tahap ini dilakukan penjumlahan masing-masing nilai *similarity* untuk mendapatkan total *similarity* [12].

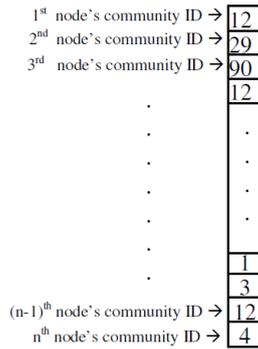
Kesamaan total antar pengguna u_i dan u_j dapat dihitung dengan rumus berikut ini :

$$\text{sim}_{\text{total}}(i,j) = \alpha \text{sim}_{\text{mention}}(i,j) + \beta \text{sim}_{\text{reply}}(i,j) + \gamma \text{sim}_{\text{follow}}(i,j) \quad (2.4)$$

Dimana nilai α , β , dan γ merupakan parameter 0 sampai 1 untuk mengontrol bobot dari setiap similarity, dan hasil $\alpha + \beta + \gamma = 1$

2.5 Algoritma Genetika

Dalam *community detection*, nilai *fitness* diambil dari nilai *similarity* yang telah dihitung sebelumnya untuk setiap anggota solusi dan terdapat modifikasi langkah-langkah dari algoritma genetika untuk memenuhi kebutuhan algoritma *community detection*. seperti perubahan dalam operasi cross-over dan penyisipan beberapa langkah tambahan selama pembuatan populasi awal. Algoritma dimulai dengan penciptaan populasi awal. *Integer array* (atau vektor) yang digunakan untuk representasi data masalah *community detection*. Penyimpanan *array identifier* (commID) node, yaitu a_i adalah identifier komunitas node i . *Array* memiliki n elemen dan disebut sebagai kromosom dalam hal algoritma genetika. Ada sejumlah kromosom memegang informasi konfigurasi komunitas yang berbeda dalam populasi. Lihat Gambar 2.4 ini untuk representasi data kromosom pada algoritma [12].



Gambar 1-1 Representasi kromosom

2.6 Network Modularity

Modularity merupakan properti yg dimiliki oleh jaringan dan dapat digunakan sebagai patokan dalam menentukan kualitas komunitas yang ada pada jaringan tersebut. Jaringan (yang berisi komunitas-komunitas yg terdeteksi) yang memiliki modularity yang tertinggi adalah yang menjadi output dari algoritma pencarian yg diterapkan di penelitian ini [1].

Rumus Modularity sebagai berikut :

$$Q = \frac{1}{2m} \sum_{vw} \left[A_{vw} - \frac{k_v k_w}{2m} \right] \delta(c_v, c_w) \quad (2.5)$$

Keterangan :

- A_{vw} merupakan bentuk Matriks Adjacency :

$$A_{vw} = \begin{cases} 1 & \text{jika vertex } v \text{ dan } w \text{ terhubung} \\ 0 & \text{jika vertex } v \text{ dan } w \text{ tidak terhubung} \end{cases}$$

- v dan w berada dalam satu komunitas jika u dan v terhubung
- c_v merupakan komunitas yang didalamnya terdapat vertex v
- c_w merupakan komunitas yang didalamnya terdapat vertex w
- Fungsi $\delta(c_v, c_w)$ akan bernilai 1 apabila $c_v = c_w$ dan 0 untuk lainnya
- $m = \frac{1}{2} \sum vw A_{vw}$ adalah jumlah edge (kuantitas) yang ada pada graf.
- Derajat k_v dari vertex v dapat didefinisikan sebagai jumlah edge yang terbentuk dari graf, dengan rumus $k_v = \sum w A_{vw}$
- Derajat k_w dari vertex w dapat didefinisikan sebagai jumlah edge yang terbentuk dari graf, dengan rumus $k_w = \sum v A_{vw}$
- Edge yang terjadi antara vertex v dan w pada koneksi yang terjadi secara acak dan tetap memperhitungkan nilai derajat vertex adalah $\frac{k_v k_w}{2m}$

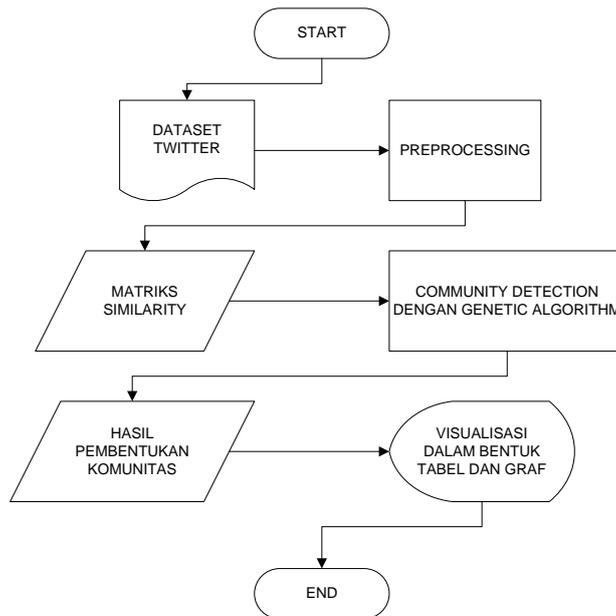
3. Perancangan dan Implementasi

3.1 Perancangan Sistem

Sistem *community detection* dengan algoritma *Genetic Algorithm* adalah sistem yang digunakan untuk membentuk struktur komunitas yang didasarkan pada *dataset* yang diambil dari *social network twitter*. Sistem ini bertujuan untuk menghitung jumlah komunitas yang terbentuk dan anggota dari masing-masing komunitas serta nilai modularity dari setiap komunitas yang didasarkan pada nilai similarity dari setiap interaksi yang dilakukan antar user. Proses pembentukan komunitas dengan algoritma *Genetic Algorithm* dilakukan dalam tahapan-tahapan sebagai berikut :

- a. Melakukan pengambilan *dataset* dari *social network twitter*.
- b. *Preprocessing* dataset *twitter*. Tahap *preprocessing* dibagi menjadi 3 tahap, yaitu :
 - *Parsing* : proses pemisahan node dan pemisahan koneksi (*following/followed, mention dan reply*). Output dari tahap ini berupa table node dan table relasi.
 - Pembuatan matriks $n \times n$: pemberian nilai dari banyaknya interaksi. Output dari tahap ini berupa matriks $n \times n$.
 - Pembuatan matriks *similarity* : perhitungan menggunakan rumus similarity setiap interaksi. Output dari tahap ini adalah matrix *similarity*.
- c. Pembentukan komunitas menggunakan Algoritma *Genetic Algorithm* berdasarkan matriks *similarity*.
- d. Menyajikan data hasil pembentukan komunitas dalam bentuk tabel.
- e. Memvisualisasikan komunitas yang telah terbentuk dalam bentuk graf.

Tahapan *community detection* dalam sistem ini secara umum digambarkan dalam *flowchart* pada Gambar 3.1.



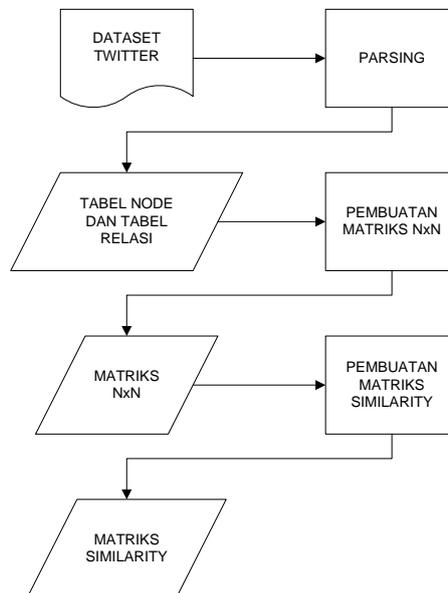
Gambar 1-2 Flowchart rancangan system

3.1.1 Dataset

Dataset yang digunakan dalam sistem ini adalah *dataset* yang diambil dari *social network twitter* dengan mengambil data pada <http://www.nodexlgraphgallery.org/>. dengan menggunakan *software* NodeXL, data dipisahkan berdasarkan nama pengguna, data relasi *following*, *follower*, *mention* dan *reply* berdasarkan tag pada file XML. Data yang digunakan terdiri dari data nama pengguna, data relasi antar pengguna berupa *following*, *follower*, *mention* dan *reply*.

3.1.2 Preprocessing

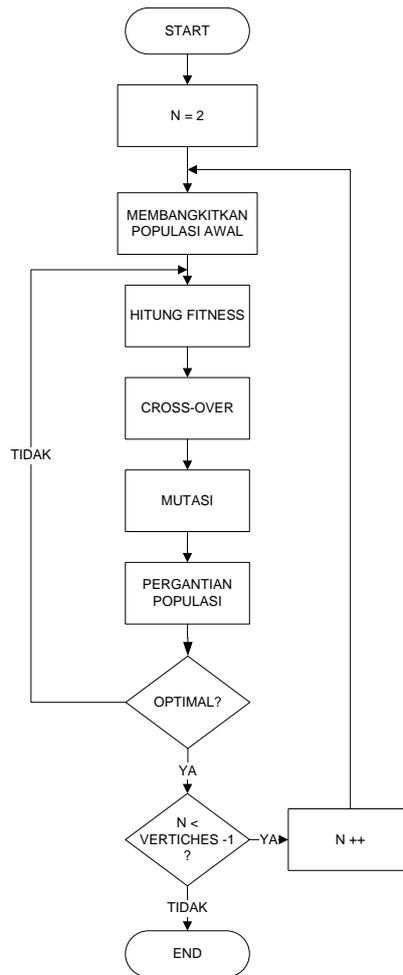
Preprocessing dilakukan dengan tujuan untuk memberikan pembobotan nilai *similarity* pada tiap relasi antar user yang ada dari *dataset twitter* yang berupa XML. Proses *preprocessing* memiliki 3 tahapan yaitu parsing, pembuatan matriks $n \times n$, dan pembentukan *similarity* matriks. Tahapan *preprocessing* dapat digambarkan dalam *flowchart* pada Gambar 2.6.



Gambar 1-3 Flowchart tahapan preprocessing

3.1.3 Algoritma Genetika

Setelah mendapatkan bobot nilai *similarity*, selanjutnya digunakan Algoritma *Genetic Algorithm* untuk proses pembentukan komunitas. Alur dalam algoritma ini secara umum dapat digambarkan dengan *flowchart* pada Gambar 3.3.



Gambar 1-4 Flowchart tahapan Genetic Algorithm

3.1.4 Network modularity

Setelah dilakukan penentuan komunitas beberapa kali menggunakan *Genetic Algorithm*, semua hasil komunitas dihitung nilai *Network Modularity* nya. Jaringan dengan *Network Modularity* paling besar akan menjadi solusi akhir dari *Community Detection*.

3.2 Implementasi

3.2.1 Kebutuhan perangkat keras

Spesifikasi perangkat keras yang digunakan untuk implementasi sistem ini adalah :

- Processor : Intel® Core™ i3 CPU M 330 @ 2.13GHz
- Memory (RAM) : 3.00 GB (2.86 GB usable)
- VGA : Intel® HD Graphics
- HDD : 250 GB

3.2.1 Kebutuhan perangkat lunak

Spesifikasi kebutuhan perangkat lunak yang digunakan untuk implementasi sistem ini adalah :

- Sistem Operasi Windows 7 Ultimate 32-bit
- Software NodeXL
- Microsoft Excel 2013
- Netbeans IDE 8.0.1

4. Pengujian dan Analisis

4.1. Pengujian

Pengujian yang dilakukan pada penelitian ini yaitu pengujian untuk mengetahui pengaruh nilai *similarity* terhadap pembentukan komunitas menggunakan *Genetic Algorithm*. Selain itu pengujian ini memiliki tujuan untuk menganalisis nilai *modularity* dari komunitas yang dihasilkan oleh *Genetic Algorithm*. Nilai *modularity* dianalisis dengan membandingkan nilai *modularity* dari setiap iterasi *Genetic Algorithm* pada setiap proses pengujian data. Pada akhir proses pencarian komunitas akan dilakukan analisis nilai *modularity* hasil penelitian dengan nilai *modularity* yang didapatkan dari software *NodeXL* yang merupakan sumber data dari penelitian ini.

4.1.1 Skenario Pengujian

Skenerio pengujian dari penelitian ini adalah :

- a. Data yang akan diuji adalah 3 data yang memiliki jumlah pengguna dan jumlah interaksi yang berbeda-beda.
- b. Ketiga data tersebut diambil dari twitter dengan menggunakan *software NodeXL*. Pengambilan data didasarkan pada pengguna dan *keyword* tertentu.
- c. Data yang didapatkan dari twitter *diparsing* oleh software *NodeXL* menjadi beberapa bagian data dalam bentuk file excel.
- d. Data yang telah *diparsing* selanjutnya akan dibaca dan dipisahkan menjadi data *Vertices* dan *Dataset* relasi.
- e. Data tersebut selanjutnya dilakukan proses *preprocessing* yaitu proses pembuatan matriks pembobotan sesuai dengan relasi yang ada. Selanjutnya matriks pembobotan tersebut dilakukan perhitungan nilai *similarity* menggunakan rumus *similarity* dan kemudian menjadi *matriks similarity*.
- f. Selanjutnya dilakukan proses pencarian komunitas menggunakan *Genetic Algorithm*. Pada proses pencarian ini akan di analisis perubahan nilai *modularity* nya pada setiap iterasi *Genetic Algorithm*. Iterasi ini dilakukan sebanyak jumlah *user – 2*, dari *maximum community* sebanyak 2 sampai *maximum community* sebanyak jumlah *user – 1*.
- g. Setelah ditemukan nilai *modularity* terbaik dan dihasilkan komunitas terbaik, dilakukan analisis nilai *modularity* akhir dengan nilai *modularity* dari komunitas yang dihasilkan oleh software *NodeXL*.

4.1.2 Proses Pengujian

Proses pengujian dibagi dalam beberapa tahap yaitu proses pengambilan data, *parsing data*, pembobotan matriks dan pembuatan matriks *similarity*, dan proses pencarian komunitas.

4.1.3 Pengambilan Data

Pengambilan data dilakukan dengan menggunakan software *NodeXL*. Dalam pengujian ini diambil 3 data dengan rincian sebagai berikut:

a. Data pertama

Keyword : -
Jumlah Pengguna : 50
Jumlah Interaksi
- Mention : 330
- Reply : 222
- Follow : 1045

b. Data kedua

Keyword : “*idul fitri*” OR lebaran OR “*hari raya*”
Jumlah Pengguna : 64
Jumlah Interaksi
- Mention : 91
- Reply : 40
- Follow : 249

c. Data ketiga

Keyword : “*Telkom university*” OR “*universitas Telkom*” OR “*Tel-U*”
Jumlah Pengguna : 100
Jumlah Interaksi
- Mention : 498
- Reply : 120
- Follow : 587

4.1.4 Parsing Data

Proses *parsing* data dilakukan dengan tujuan untuk memisahkan data-data dari dataset menjadi data yang siap untuk diolah dalam proses pembobotan matriks. Data dipisahkan menjadi 2 bagian yaitu *dataset relasi* dan data *vertices*. *Dataset relasi* adalah data yang berisi relasi antar user, sementara *vertices* merupakan data semua user yang ada.

4.1.5 Pembobotan matriks dan pembuatan matriks similarity

Pembobotan matriks bertujuan untuk membuat matriks yang berisi hubungan antar user berdasarkan relasi *mention*, *reply*, *follower* dan *followed*. Kemudian dari setiap matriks tersebut dihitung nilai *similarity*nya untuk kemudian dijadikan *matriks similarity mention*, *reply*, dan *follow*. Kemudian dari ketiga matriks tersebut dijumlahkan sehingga menghasilkan matriks *similarity* total.

4.1.6 Pencarian komunitas menggunakan genetic Algorithm

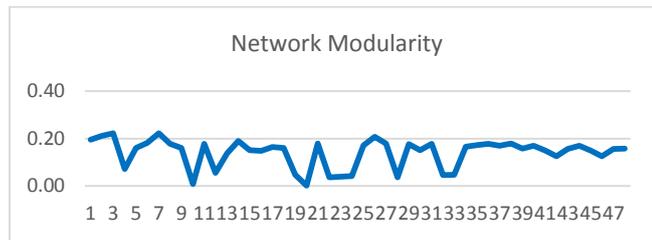
Setelah matriks *similarity* terbentuk, dilakukan proses pencarian komunitas yang didasarkan pada matriks *similarity* tersebut. Proses pencarian komunitas dilakukan dengan menggunakan *Genetic Algoritma* yang dilakukan beberapa kali yang kemudian dipilih nilai *network modularity* yang terbesar untuk dijadikan solusi.

4.2 Analisis Perubahan Nilai Modularity setiap iterasi GA

Perubahan nilai *modularity* pada setiap iterasi *Genetic Algorithm* dianalisis untuk mengetahui perbedaan nilai *similarity* pada setiap pembatasan nilai maksimal komunitas pada setiap iterasi.

a. Data pertama

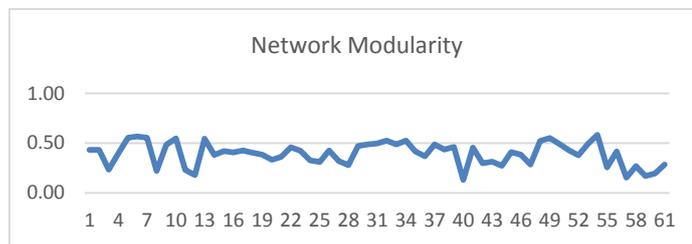
Grafik 4-1 Grafik perubahan nilai modularity pada data pertama



Grafik 4-2 Grafik perubahan nilai modularity pada data pertama percobaan kedua

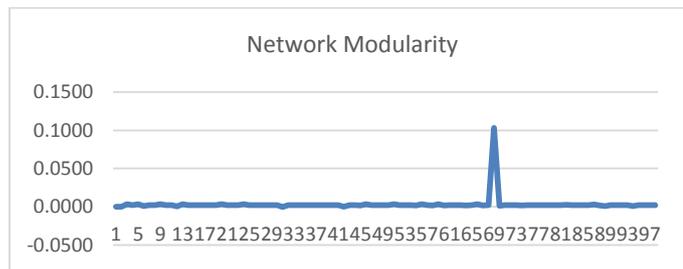
b. Data kedua

Grafik 4-3 Grafik perubahan nilai modularity pada data kedua



c. Data ketiga

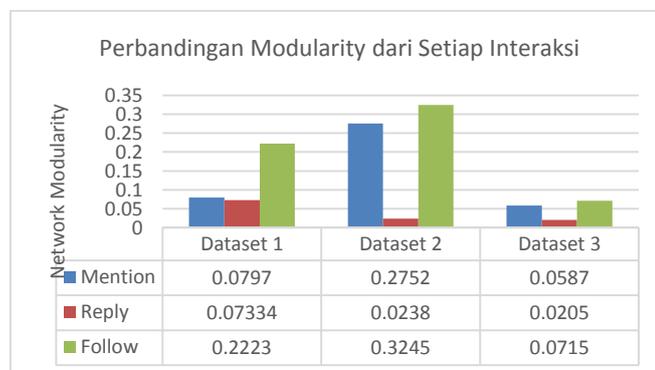
Grafik 4-4 Grafik perubahan nilai modularity pada data ketiga



4.3 Analisis nilai modularity pada setiap interaksi

Dilakukan analisis terhadap nilai modularity yang didapatkan dari proses community detection dengan data dari masing-masing similarity interaksi yaitu similarity *mention*, *reply*, dan *follow*.

Grafik 4-5 Perbandingan Modularity dari Setiap Interaksi

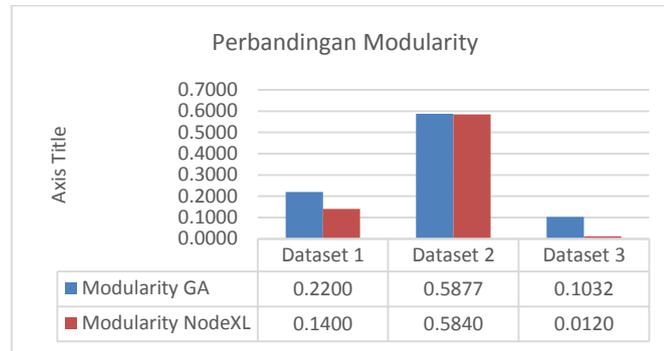


Grafik 4-5 menunjukkan perbandingan nilai *modularity* dari setiap interaksi dimana relasi *follow* pada ketiga data merupakan interaksi dengan nilai *modularity* terbesar sementara nilai terbesar kedua adalah relasi *mention*.

4.4 Analisis nilai modularity hasil Community Detection

Komunitas hasil pencarian dengan *Genetic Algorithm* akan dianalisis nilai *modularity*nya dengan cara dibandingkan dengan hasil komunitas yang dihasilkan oleh *software* NodeXL. Berikut adalah hasil analisis dari setiap dataset yang ada.

Grafik 4-6 Perbandingan Modularity



Grafik 4-6 menunjukkan perbandingan nilai *modularity* hasil pembentukan komunitas dengan menggunakan *Genetic Algorithm* dan Algoritma Girvan-Newman. Dapat dilihat bahwa nilai *modularity* dengan menggunakan *Genetic Algorithm* pada ketiga data lebih baik daripada nilai *modularity* dengan menggunakan Algoritma Girvan-Newman.

5. Penutup

5.1. Kesimpulan

Berdasarkan proses pengujian dan analisis yang telah dilakukan, dapat ditarik kesimpulan sebagai berikut :

- Pada proses pencarian komunitas, pengaruh interaksi *mention*, *reply* dan *follow* antar user telah di analisis dengan hasil interaksi *follow* menjadi interaksi yang paling berpengaruh dalam pembentukan komunitas. Namun ketiga interaksi memiliki kontribusi masing-masing dalam proses pembentukan komunitas sehingga dapat memaksimalkan nilai *modularity*.
- Proses pembentukan komunitas yang dibentuk menggunakan *Genetic Algorithm* dilakukan dalam beberapa iterasi untuk memaksimalkan nilai *network modularity* sehingga komunitas yang terbaik didapatkan.
- Nilai *modularity* dari pembentukan komunitas menggunakan algoritma yang didasarkan pada *Genetic Algorithm* sesuai dengan percobaan yang telah dilakukan sebanyak 3 kali, menunjukkan bahwa nilai *modularity* yang dihasilkan oleh *Genetic Algorithm* lebih baik daripada nilai *modularity* yang dihasilkan oleh *software* NodeXL dengan algoritma *Girvan-Newman*.

5.2. Saran

Adapun saran perbaikan yang dapat dilakukan untuk penelitian selanjutnya adalah sebagai berikut :

- Data yang dipakai sebaiknya memiliki interaksi yang lebih banyak lagi dan beragam sehingga komunitas yang dibentuk lebih kompleks.
- Perlunya data acuan yang telah teruji keakuratannya untuk dijadikan data uji penelitian dan sebagai acuan analisis data.
- Mengembangkan penelitian pencarian komunitas menggunakan data pada social media lainnya seperti *Facebook*, *Path*, *Instagram*, dan lain sebagainya.

6. Daftar Pustaka

- [1] Clauset, Aaron dkk. *Finding community structure in very large networks*. University of New Mexico. 2010
- [2] Duch, J., Arenas, A. "Community detection in complex networks using extremal optimization", Pre-print condmat/0501368, 2005.
- [3] Girvan, M., Newman, M.E.J.: *Community structure in social and biological networks*. Proc. National. Academy of Science. USA 99, 7821–7826 (2002)
- [4] Kewalramani, Mohit Naresh. *Community Detection In Twitter*. University of Maryland. 2011.
- [5] Miller, Claire Cain (October 30, 2010). *Why Twitter's C.E.O. Demoted Himself*. The New York Times. Diakses October 31, 2010
- [6] Newman, M.E.J., Girvan, M. "Finding and evaluating community structure in networks", Physical Review E, 69:026113, 2004.
- [7] Pan, Long. *Effective and Efficient Methodologies for Social Network Analysis*, State University. 2012
- [8] Qi, Go-jun et. al. *Community Detection with Edge Content in Social Media Networks*. University of Illinois and IBM T.J. Watson Research Center.
- [9] Tangsin, Mursel and Bingol, Haluk. *Community Detection in Complex Networks using Genetic Algorithm*. Bogazici University, Istanbul, Turkey. 2007.
- [10] Utami, Nisa Citra. *Analisis dan Implementasi Community detection Menggunakan Algoritma Girvan and Newman dalam Social Network*. IT Telkom. 2014
- [11] Wasserman, Stanley; Faust, Katherine (1994). *Social Network Analysis in the Social and Behavioral Sciences*. Social Network Analysis: Methods and Applications. Cambridge University.
- [12] Yang, Zhang, et. al. *Community Discovery in Twitter Based on User Interests*. National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China. 2012.