

ANALISIS DAN IMPLEMENTASI ALGORITMA DYNAMICNET PADA DETEKSI EVOLUSI KOMUNITAS DI MEDIA SOSIAL TWITTER

Muhammad Rizky Riandi Gunaedi¹, Imelda Atastina, S.I., M.T.², Anisa Herdiani, S.T., M.T.³

^{1,2,3}Fakultas Informatika, Universitas Telkom, Bandung

¹rizkyriandi@students.telkomuniversity.ac.id, ²imelda@telkomuniversity.ac.id,

³anisaherdiani@telkomuniversity.com

Abstrak

Seiring berkembangnya jaringan sosial, banyak media-media di internet yang menyediakan sarana untuk berhubungan dengan orang lain, salah satunya adalah Twitter. Twitter memungkinkan terbentuknya sebuah komunitas. Komunitas berkembang setiap saat, seiring dengan berkembangnya interaksi antar pengguna di Twitter, oleh karena itu diperlukan alat untuk mendeteksi evolusi dari komunitas tersebut. DynamicNet merupakan algoritma untuk mendeteksi evolusi komunitas yang mendefinisikan evolusi komunitas dengan simpel namun mencakup banyak bidang, yang mana sebelum mendeteksi evolusi dilakukan deteksi komunitas menggunakan algoritma Louvain. Pada jurnal ini dilakukan percobaan untuk mengetahui parameter apa sajakah yang mempengaruhi hasil dari algoritma DynamicNet, mulai dari jumlah data dan threshold. Hasil yang didapat dari penelitian ini adalah algoritma DynamicNet dapat digunakan untuk mendeteksi evolusi komunitas di media sosial Twitter. Didapatkan juga bentuk kerasteristik data yang ideal agar algoritma DynamicNet dapat berjalan dengan optimal, yaitu jumlah data yang banyak dengan kepadatan hubungan yang tinggi, komunitas dengan kualitas (Modularity) yang tinggi dan nilai threshold yang berada di rentan nilai *Normalized Mutual Information* (NMI). Pada penelitian ini NMI tertinggi dicapai pada 1000 data simpul yang bernilai 0.16.

Kata kunci: *dynamic network, dynamicnet, twitter, community evolution, nmi*

Abstract

As the development of social networks, many media on the internet that provides the means to connect with others, one of which is Twitter. Twitter wants to form a community. Communities are evolving all the time, along with the growing interaction between users on Twitter, therefore a tool for the evolutionary part of the community is needed. DynamicNet is an algorithm for. Which uses Louvain's algorithm. In this journal an experiment was conducted to find out what parameters are the results of the DynamicNet algorithm, ranging from the amount of data and threshold. The result of this research is DynamicNet algorithm can be used for. The ideal data rectangle shape for DynamicNet algorithm can be run optimally, ie high amount of data with high link density, high quality (Modularity) communities and Dynamic Mutual Dynamic Information (NMI) threatened threshold values. In tis research, the highest NMI value archive with 1500 vertex data which has value 0.0157.

Keyword: *dynamic network, dynamicnet, twitter, community evolution, nmi*

1. Pendahuluan

1.1. Latar Belakang

Belakangan ini, ketertarikan akan jaringan informasi mulai banyak bermunculan, rata rata dilatar belakangnya oleh banyaknya masalah dalam penelitian jaringan informasi, seperti representasi, permodelan, penambahan data jaringan informasi dalam jumlah besar, serta media sosial dan jaringan sosial [1]. Seiring pertumbuhan internet, Web 2.0 dan media sosial, menimbulkan tantangan baru untuk menganalisis jaringan sosial dalam skala yang lebih besar. Interaksi sosial pada media sosial dapat berupa, komunikasi email, *instant messenger*, hubungan telepon, dan hubungan antar teman [2]. Twitter merupakan media sosial *micro-blogging* yang dibatasi hanya 140 karakter per *post*. Twitter memiliki 313 juta pengguna aktif pertahun terhitung pada tahun 2017. Banyak hal yang dapat dilakukan pada media sosial Twitter, seperti promosi produk, media kampanye dan menjalin pertemanan antar pengguna [3]. Hal ini memicu terbentuknya sebuah komunitas dalam ruang lingkup media sosial Twitter.

Identifikasi komunitas sangat penting karena dapat membantu memvisualisasikan struktur jaringan sosial [4]. Komunitas sendiri mengalami perubahan seiring berjalannya waktu, komunitas dapat berkembang, menyusut, lahir dan bahkan mati [2] [1]. Untuk mendeteksi evolusi komunitas terdapat beberapa algoritma, seperti Algoritma DynamicNet, Algoritma deteksi evolusi berdasarkan *Incremental Indetification* [5] dan Algoritma deteksi evolusi berdasarkan Partikel dan Kepadatan [6]. Algoritma DynamicNet adalah sebuah algoritma mendeteksi evolusi komunitas dimana untuk mendeteksi komunitasnya menggunakan algoritma Louvain, selanjutnya hasil algoritma Louvain akan dibandingkan menggunakan DynamicNet untuk mengetahui evolusi

dari komunitas [1]. Algoritma DynamicNet sendiri dipilih karena memiliki pendefinisian proses evolusi komunitas yang lebih simpel namun mencakup banyak proses dibandingkan algoritma yang lain. Pada penelitian ini ingin dibuktikan kemampuan DynamicNet dalam mendeteksi evolusi komunitas pada media sosial Twitter. Kedepannya hasil dari penelitian ini dapat digunakan salah satunya sebagai bahan untuk pengambil keputusan suatu perusahaan, seperti memutuskan untuk beriklan di Twitter atau tidak.

1.2. Perumusan Masalah

Berdasarkan latar belakang yang sudah dipaparkan di atas, dapat ditarik perumusan masalah untuk penelitian ini yaitu bagaimana proses mendeteksi evolusi komunitas pada data yang berasal dari media sosial Twitter menggunakan algoritma DynamicNet. Seperti yang diketahui jika algoritma DynamicNet menggunakan algoritma Louvain untuk melakukan proses deteksi komunitas, oleh karena itu perlu diketahui parameter apa saja yang mempengaruhi terbentuknya komunitas oleh algoritma Louvain dan nilai modularity dari komunitas yang terbentuk. Selain untuk mengetahui parameter yang mempengaruhi algoritma Louvain juga perlu diketahui parameter yang mempengaruhi hasil dari algoritma DynamicNet, adapun untuk menguji kualitas data yang digunakan algoritma DynamicNet akan dilihat menggunakan nilai NMI. Beberapa kandidat parameter untuk algoritma Louvain adalah jumlah data, jumlah sisi, dan nilai $\alpha\beta\gamma$, sedangkan untuk algoritma DynamicNet kandidatnya adalah jumlah data, jumlah sisi, nilai $\alpha\beta\gamma$ dan nilai threshold. Batasan dalam penelitian ini adalah data graf yang digunakan hanya graf yang tidak berarah yang berbobot. Relasi antar pengguna Twitter yang digunakan dalam penelitian ini hanya *Reply*, *Mention* dan *Follow* karena ketiga relasi tersebut merupakan relasi dasar dari interaksi pada media sosial Twitter sehingga isi *tweet* tidak diproses dalam penelitian ini. Hasil komunitas yang diperoleh juga tidak menjelaskan penentuan topik dan nama dari komunitas tersebut.

1.3 Tujuan

Tujuan dari penelitian ini adalah untuk mengetahui proses mendeteksi evolusi komunitas pada data yang berasal dari media sosial Twitter menggunakan algoritma DynamicNet juga untuk mengetahui parameter apa saja yang mempengaruhi hasil dari algoritma DynamicNet dan Louvain. Penelitian ini dilakukan juga untuk mengetahui performansi terbaik dari algoritma DynamicNet berdasarkan parameter yang diuji.

1.4 Organisasi Tulisan

Penulisan Tugas Akhir ini disusun dalam beberapa bagian, yaitu: Bagian 1 – Pendahuluan, Bagian 2 – Studi Pustaka, Bagian 3 – Sistem yang Dibangun, Bagian 4 – Evaluasi, Bagian 5 – Kesimpulan dan Saran.

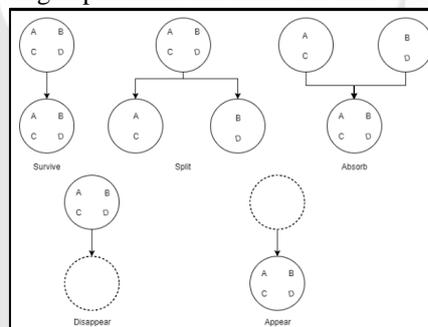
2. Studi Pustaka

2.1. Algoritma Louvain

Komunitas dapat di sebut juga sebagai *cluster*, sebuah kumpulan simpul yang memiliki ketertarikan yang sama di dalam jaringan. Deteksi komunitas sendiri dapat diartikan sebagai pembagian simpul pada sebuah jaringan menjadi beberapa kelompok, yang mana memaksimalkan jumlah sisi didalam kelompok dan meminimalisasi jumlah sisi antar simpul pada grup yang berbeda [7]. Algoritma Louvain adalah salah satu dari banyak algoritma untuk mendeteksi komunitas. Salah satu kelebihan dari algoritma Louvain adalah mendeteksi komunitas dengan *modularity* yang maksimal dan juga lebih cepat jika dibandingkan dengan algoritma lain. Cara kerja dari algoritma Louvain sendiri dibagi menjadi 3 fase yang dapat dilihat pada paper [4].

2.2. Algoritma DynamicNet

Pada umumnya, jaringan sosial menunjukkan hubungan antar individu, jaringan sosial juga dapat berubah seiring berjalannya waktu, sebagai contoh: hubungan persahabatan antar dua individu [8]



Gambar 1. Ilustrasi Evolusi Komunitas [1]

Perubahan pada jaringan sosial tentu memberikan efek pada pembentukan komunitas di waktu yang berbeda, karena komunitas sendiri mengalami perubahan seperti: *survive*, *absorb*, *split*, *disappear* dan *appear* seperti pada Gambar 1 [1]. Mendeteksi pola interaksi dan memprediksi struktur dari jaringan di masa depan sendiri memiliki beberapa manfaat, seperti dalam bidang marketing, maksimalisasi *revenue*, dan juga dapat membantu keputusan

strategi marketing [8]. Algoritma DynamicNet adalah sebuah algoritma yang di rancang oleh Alfredo Cuzzocrea, Francesco Folino, dan Clara Pizzuti dimana algoritma DynamicNet digunakan untuk mendeteksi evolusi dari sebuah komunitas. DynamicNet mendeskripsikan evolusi komunitas menjadi 5 bagian yaitu: *survive*, *absorb*, *split*, *disappear* dan *appear*. Adapun cara kerja dari algoritma DynamicNet dapat dilihat pada paper [1].

2.3. Similarity Measurement

Similarity adalah tingkat kesamaan antar pengguna berdasarkan banyaknya interaksi antar pengguna Twitter menggunakan interaksi *follow*, *mention* dan *reply*, nantinya nilai *similarity* setiap relasi tersebut akan di jumlahkan dan akan di jadikan *similarity* total [9]. Adapun perhitungan masing-masing *similarity* adalah sebagai berikut:

1. Similarity Mention

Tujuan perhitungan *similarity* terhadap relasi *mention* adalah untuk melihat seberapa sering 2 orang pengguna melakukan *mention*. Yang mana dalam perhitungannya jumlah pengguna yang di *mention* oleh U_i dan U_j didefinisikan sebagai $C_{mention}$, $|R_i|$ adalah jumlah pengguna yang di *mention* oleh U_i , $|R_j|$ adalah jumlah pengguna yang di *mention* oleh U_j , n_i adalah jumlah pengguna U_j di *mention* oleh U_i dan n_j merupakan jumlah pengguna U_i di *mention* oleh U_j . Adapun untuk menghitung *similarity mention* dapat digunakan rumus berikut [9].

$$sim_{mention} = \frac{C_{mention}}{\sqrt{|R_i|}\sqrt{|R_j|}} + \frac{n_i+n_j}{|R_i||R_j|} \quad (2)$$

2. Similarity Reply

Tujuan perhitungan *similarity* terhadap relasi *reply* bertujuan untuk melihat seberapa sering *reply* dilakukan antar 2 orang pengguna. Adapun untuk menghitung *similarity reply* dapat digunakan rumus berikut [9].

$$sim_{reply} = \frac{C_{reply}}{\sqrt{|R_i|}\sqrt{|R_j|}} + \frac{n_i+n_j}{|R_i||R_j|} \quad (3)$$

Similarity reply menghitung jumlah pengguna yang di *reply* oleh U_i dan U_j didefinisikan sebagai C_{reply} , $|R_i|$ (jumlah pengguna yang di *reply* oleh U_i), $|R_j|$ (jumlah pengguna yang di *reply* oleh U_j), n_i (jumlah pengguna U_j di *reply* oleh U_i) dan n_j = jumlah pengguna U_i di *reply* oleh U_j .

3. Similarity Follow

Tujuan perhitungan *similarity* terhadap relasi *follow* bertujuan untuk melihat seberapa banyak *following/follower* yang sama antara 2 pengguna dengan menghitung jumlah *following* yang sama dari U_i dan U_j yang didefinisikan sebagai C_{friend} , adalah jumlah *follower* yang sama dari U_i dan U_j ($C_{follower}$), $|friend_i|$ (jumlah pengguna yang di *follow* oleh U_i), $|friend_j|$ (jumlah pengguna yang di *follow* oleh U_j), $|follower_i|$ (jumlah *follower* dari U_i) dan $|follower_j|$ (jumlah *follower* dari U_j). Adapun untuk menghitung *similarity mention* dapat digunakan rumus (4) berikut [9].

$$sim_{follow} = \frac{C_{friend}}{\sqrt{|friend_i|}\sqrt{|friend_j|}} + \frac{C_{follower}}{\sqrt{|follower_i|}\sqrt{|follower_j|}} \quad (4)$$

4. Similarity Total

Dari ketiga perhitungan *similarity* di atas dihitung nilai *similarity* total antar 2 pengguna. Adapun perhitungan *similarity* total dapat menggunakan rumus (5) berikut [9]. Nilai α , β , γ merupakan parameter antara 0 sampai 1 untuk mengontrol bobot dari setiap *similarity*. Dimana apabila $\alpha + \beta + \gamma = 1$.

$$sim_{total}(i,j) = \alpha sim_{mention}(i,j) + \beta sim_{reply}(i,j) + \gamma sim_{follow}(i,j) \quad (5)$$

2.4. Modularity

Modularity adalah sebuah alat ukur yang ditemukan oleh Newman yang digunakan untuk mengukur kualitas dari sebuah komunitas pada algoritma deteksi komunitas [10]. Nilai modularity sendiri terbatas dari -1 hingga 1 dimana nilai tersebut mengukur kepadatan hubungan didalam sebuah komunitas yang dibandingkan dengan hubungan antara komunitas. Berikut adalah rumus dari modularity [4]:

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(C_i, C_j) \quad (6)$$

Yang mana m adalah total sisi yang ada didalam graf/jaringan, A adalah Nilai relasi antara 2 simpul yang diteliti, dimana akan bernilai 1 apabila simpul tersebut memiliki hubungan dan akan bernilai 0 jika tidak memiliki

hubungan, k adalah jumlah derajat yang dimiliki oleh simpul tersebut dan $\delta(C_i, C_j)$ merupakan fungsi δ akan bernilai 1 apabila $C_i = C_j$ dan akan bernilai 0 jika sebaliknya.

2.5. Normalized Mutual Information (NMI)

NMI atau *Normalized Mutual Information* sudah banyak digunakan sebagai acuan dalam mengukur akurasi dari sebuah algoritma deteksi evolusi komunitas. Pada dasarnya *Mutual Information* (MI) adalah bagaimana mengukur tingkat ketergantungan antara dua variabel atau secara umumnya tingkat banyaknya kesamaan informasi pada dua variabel [11]. Nilai NMI sendiri memiliki batas dari 0 hingga 1, dimana jika nilai NMI bernilai 0 maka kedua variabel tersebut bersifat independen dan jika bernilai 1 maka kedua variabel tersebut merupakan variabel dengan isi yang sama. Adapun dalam menghitung NMI, dapat menggunakan persamaan (7) seperti berikut [12]:

$$I(A, B) = \frac{-2 \sum_{i=1}^{CA} \sum_{j=1}^{CB} N_{ij} \log\left(\frac{N_{ij} \times N}{N_i \times N_j}\right)}{\sum_{i=1}^{CA} N_i \log\left(\frac{N_i}{N}\right) + \sum_{j=1}^{CB} N_j \log\left(\frac{N_j}{N}\right)} \quad (7)$$

N adalah jumlah simpul keseluruhan, N_{ij} adalah jumlah simpul di komunitas i yang berada pada komunitas j , N_i adalah jumlah simpul di komunitas i , N_j adalah jumlah simpul di komunitas j , CA adalah jumlah komunitas pada Graf A dan CB merupakan jumlah komunitas pada Graf B.

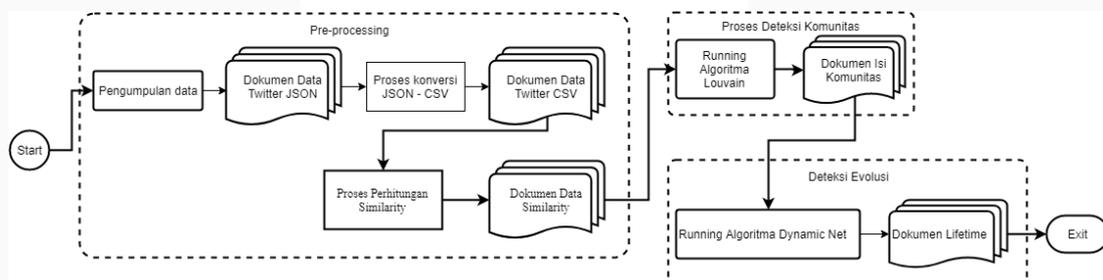
2.6 Density

Density dalam dapat diartikan sebagai kepadatan dan jika digunakan dalam graf, maka berarti menghitung kepadatan sisi dari suatu graf dengan memperhatikan jumlah sisi pada graf (E) dan jumlah simpul pada graf (V). Nilai dari Density berada antara 0 hingga 1, bernilai 0 berarti graf tersebut tidak memiliki relasi didalamnya dan jika bernilai 1 berarti setiap simpul saling terhubung. Density dapat dihitung menggunakan persamaan dibawah ini [13]:

$$d = \frac{2E}{|V|(|V|-1)} \quad (8)$$

3. Sistem yang Dibangun

Pada penelitian ini akan dibangun sistem untuk mendeteksi evolusi komunitas pada media sosial Twitter adapun keluaran dari sistem ini adalah dokumen yang berisi proses evolusi dari komunitas yang terdapat pada media sosial Twitter. Dibawah ini dilampirkan *flowchart* dari metodologi yang digunakan dalam penyelesaian sistem ini.



Gambar 2. Flowchart Metodologi Penyelesaian Sistem

Pada Gambar 2 terdapat 2 proses utama yaitu pre-processing dan deteksi evolusi komunitas. Pada proses *pre-processing*, data Twitter yang diambil menggunakan API berdasarkan *query* pencarian “Donald Trump” OR “Coca Cola” OR “Palestine” awalnya berbentuk JSON. Dalam penelitian ini data yang digunakan adalah data id sumber, id target, username source, username target, dan relasi antar keduanya. Untuk memenuhi kebutuhan data pada system, format JSON di konversi menjadi bentuk CSV dan mengeliminasi data menjadi data CSV yang berisikan *source-target-relation*, setelah data berbentuk CSV kemudian akan dihitung *similarity* antar pengguna terlebih dahulu menggunakan persamaan (2), (3), (4), (5). Setelah proses pembuatan matriks *similarity* selesai. Hasil dari proses tersebut akan menjadi masukan pada proses pendeteksian komunitas menggunakan algoritma Louvain. Cara kerja dari algoritma Louvain yang dibagi menjadi 3 fase yaitu Inisiasi, Fase1 dan Fase 2, adapun untuk penjelasan dari tiap fase dapat dilihat pada bagian Studi Pustaka pada bagian 2.1. Selanjutnya dilakukan pendeteksian evolusi dari komunitas yang sudah dideteksi pada proses Deteksi Komunitas menggunakan algoritma DynamicNet. Adapun penjelasan dari setiap proses dari algoritma DynamicNet dapat dilihat pada bagian Studi Pustaka 2.2.

4. Evaluasi

4.1. Karakteristik Data

Pada penelitian ini data yang digunakan adalah data hasil *crawling* dari media sosial Twitter, menggunakan API. Data yang digunakan dibagi menjadi 4 pembagian waktu. Rentan waktu untuk setiap data Timestep adalah 3 hari. Jumlah data simpul tiap Timestep dibagi menjadi 50, 100, 1000 dan 1500 dengan tujuan dapat melihat ketika terjadi sedikit perubahan jumlah data (50 – 100, 1000 – 1500) dan ketika terjadi perubahan jumlah data yang drastis (100 – 1000), adapun karakteristik dari data yang digunakan adalah seperti pada tabel 1:

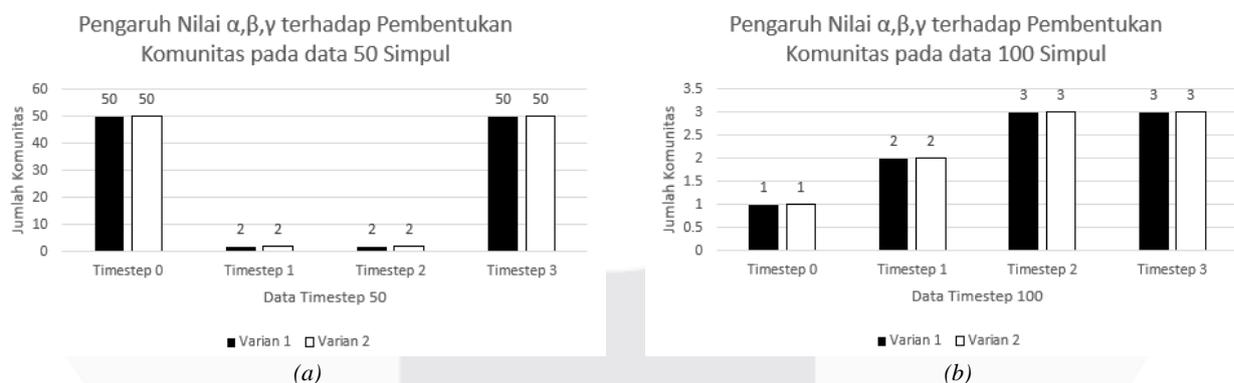
Tabel 1. Karakteristik data

Timestep	Jumlah Simpul	Jumlah Sisi	Jumlah Follow	Jumlah Mentions	Jumlah Reply	Density
0	50	11	10	1	0	0.008979592
	100	51	50	1	0	0.01030303
	1000	7405	6703	673	29	0.014824825
	1500	9355	8816	495	44	0.008321103
1	50	19	18	0	1	0.015510204
	100	106	106	0	0	0.021414141
	1000	11536	10833	676	27	0.023095095
	1500	10193	9588	582	44	0.009066489
2	50	10	10	0	0	0.008163265
	100	62	61	1	0	0.012525253
	1000	16713	16054	629	30	0.033459459
	1500	4500	4357	131	12	0.004002668
3	50	39	37	2	0	0.031836735
	100	114	110	4	0	0.023030303
	1000	13282	12587	672	23	0.026590591
	1500	2897	2711	175	11	0.002576829

Pada Tabel 1 terdapat bagan Density yang berisikan informasi tentang kepadatan graf dari Timestep tersebut. Perhitungan Density sendiri dilakukan menggunakan persamaan (8) yang dijelaskan pada bagian 2.6. Pada Tabel 1 juga diberikan penjelasan tentang rincian dari data yang digunakan, seperti jumlah sisi pada tiap data simpul, dan jumlah relasi untuk setiap data simpulnya.

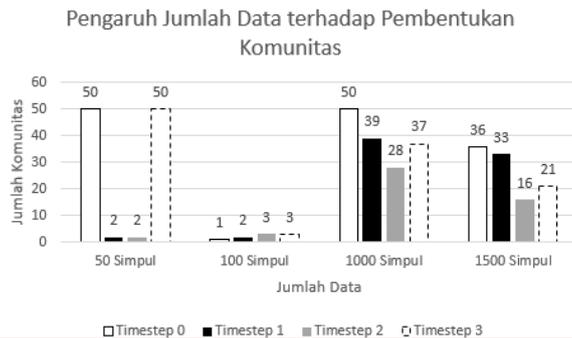
4.1.1. Analisis Hasil Skenario Pengujian Louvain

Pengujian yang dilakukan bertujuan untuk mengetahui parameter yang mempengaruhi hasil algoritma Louvain dalam pembentukan komunitas berdasarkan beberapa parameter yaitu nilai α , β , γ , jumlah data,. Kemudian dengan memperhatikan kepadatan graf akan dilihat hasil perhitungan modularity dan waktu olah algoritma Louvain untuk mengetahui performansi dari algoritma Louvain. Pengujian dilakukan menggunakan 4 ukuran data simpul yang berbeda dengan 4 data Timestep yang memiliki data sisi yang berbeda dan dengan 2 nilai α , β , γ yang berbeda yaitu 0.33-0.33-0.33 dan nilai persen dari setiap kategori relasi (*follow-mention-reply*) yang terdapat dalam data tersebut. Nilai 0.33 sendiri dipilih karena merupakan nilai paling minimum untuk ketiganya dan nilai berdasarkan persen dikarenakan untuk melihat persebaran relasi yang terjadi. Berikut adalah analisis hasil skenario pengujian Louvain.



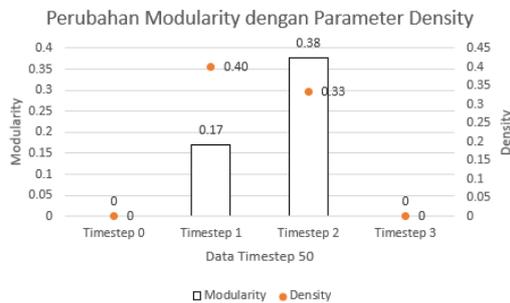
Gambar 3 Grafik Pengaruh Nilai α, β, γ terhadap Komunitas Yang Terbentuk (a) 50 Simpul, (b) 100 Simpul

Pada gambar 3 di atas membuktikan bahwa nilai α , β , γ tidak mempengaruhi proses pembentukan komunitas. Hal ini disebabkan karena nilai α , β , γ hanya mempengaruhi bobot sisi antar simpul dimana hal tersebut berarti kedua simpul memiliki relasi, sehingga berapapun nilai α , β , γ akan tetap menghasilkan bobot. Hal ini juga terjadi pada hasil pengujian menggunakan data 1000 simpul dan 1500 simpul seperti yang dapat dilihat pada Lampiran 1.

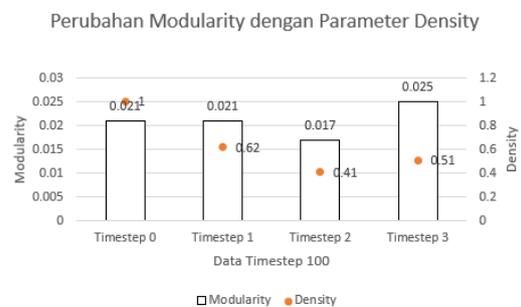


Gambar 4. Grafik Pengaruh Jumlah data terhadap Pembentukan Komunitas

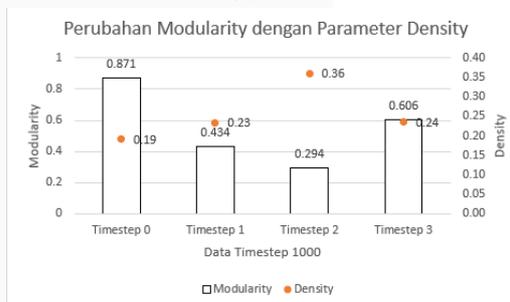
Pada Gambar 4 menunjukkan perubahan jumlah komunitas yang terbentuk dimana hal ini berarti jika jumlah data mempengaruhi dari jumlah komunitas yang dibentuk. Hal ini dapat terjadi karena semakin banyak jumlah data yang digunakan maka semakin besar kemungkinan terbentuknya komunitas lain. Pada Timestep 0 dan Timestep 3 pada 50 Simpul menunjukkan jika jumlah komunitas yang terbentuk sama dengan jumlah simpul yang digunakan, hal ini terjadi karena antar simpul tidak memiliki bobot dari hasil perhitungan similarity dengan kata lain tidak memiliki relasi, sehingga pada proses pendeteksian komunitas setiap simpul menjadi 1 komunitas. Dari kejadian tersebut dapat ditarik kesimpulan jika jumlah data menentukan nilai similarity sehingga mempengaruhi jumlah komunitas yang terbentuk.



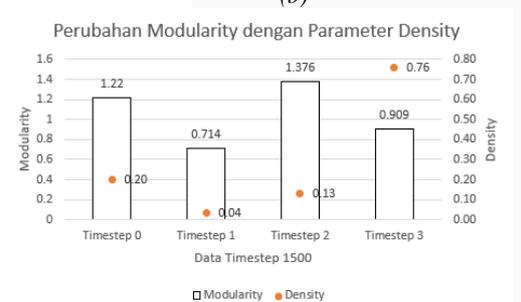
(a)



(b)



(c)



(d)

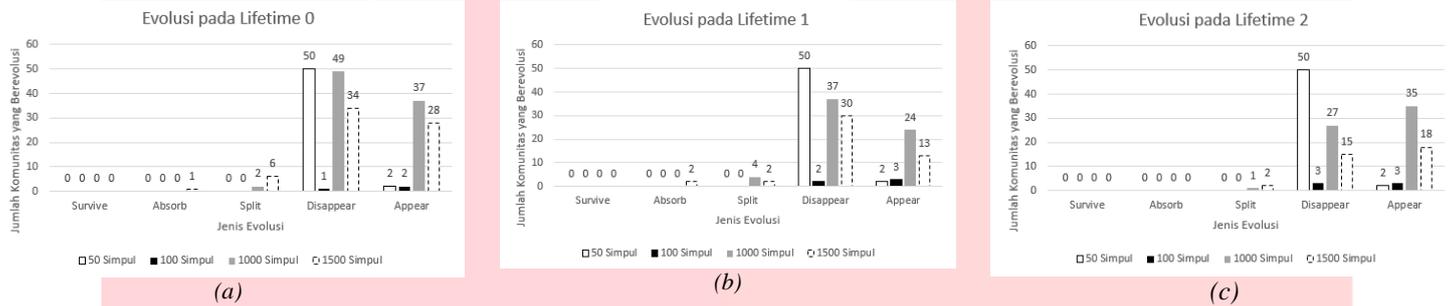
Gambar 5. grafik perubahan modularity terhadap density (a) 50 simpul, (b) 100 simpul, (c) 1000 simpul, (d) 1500 simpul

Gambar 5 di atas merepresentasikan perubahan nilai modularity terhadap kepadatan graf (density) dengan keempat data Timestep dan keempat jumlah data simpul. Dari Gambar 5 dapat diketahui jika jumlah data mempengaruhi nilai modularity dan kepadatan graf yang rinciannya dapat dilihat pada tabel 1 dan Lampiran 2(A, B, C, D). Dapat dilihat terjadi peningkatan dan penurunan modularity jika ada perubahan dari nilai kepadatan grafnya, hal ini dapat terjadi karena walaupun beberapa komunitas memiliki modularity yang besar namun komunitas yang lain bernilai 0 atau sangat kecil, sehingga wajar jika walaupun terjadi kenaikan nilai density dapat terjadi kenaikan atau penurunan jumlah nilai modularity. Perubahan yang paling wajar terdapat pada Gambar 5(b) yang menunjukkan jika semakin besar nilai density nya maka semakin tinggi nilai modularitynya dan semakin kecil nilai densitynya maka semakin kecil nilai modularitynya.

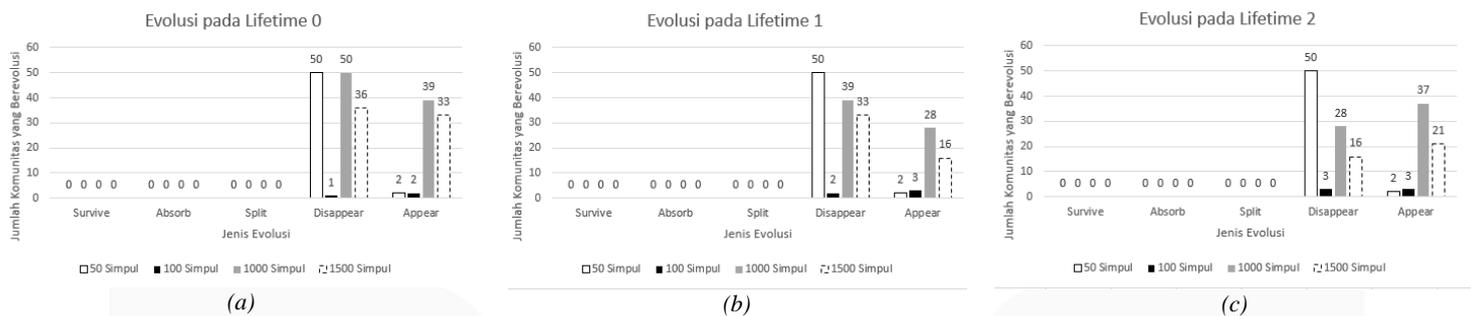
4.1.2. Analisis Hasil Skenario Pengujian DynamicNet

Pengujian yang dilakukan pada penelitian ini bertujuan untuk mengetahui parameter yang mempengaruhi hasil algoritma DynamicNet dalam mendeteksi evolusi komunitas berdasarkan jumlah data dan threshold, untuk mengetahui waktu olah algoritma DynamicNet dan untuk mengetahui kualitas data uji pada algoritma

DynamicNet, menggunakan nilai NMI. Pengujian dilakukan 2 kali menggunakan hasil implementasi Louvain yaitu berupa komunitas dan juga karakteristik data yang sama pada analisis di atas untuk mendeteksi evolusi dari komunitas tersebut dengan nilai threshold 0.5 dan threshold 0.001.



Gambar 7. Grafik Hasil Deteksi Evolusi menggunakan nilai threshold 0.001 (a) lifetime 0, (b) lifetime 1, (c) lifetime 2



Gambar 8. Grafik Hasil Deteksi Evolusi menggunakan nilai threshold 0.5 (a) lifetime 0, (b) lifetime 1, (c) lifetime 2

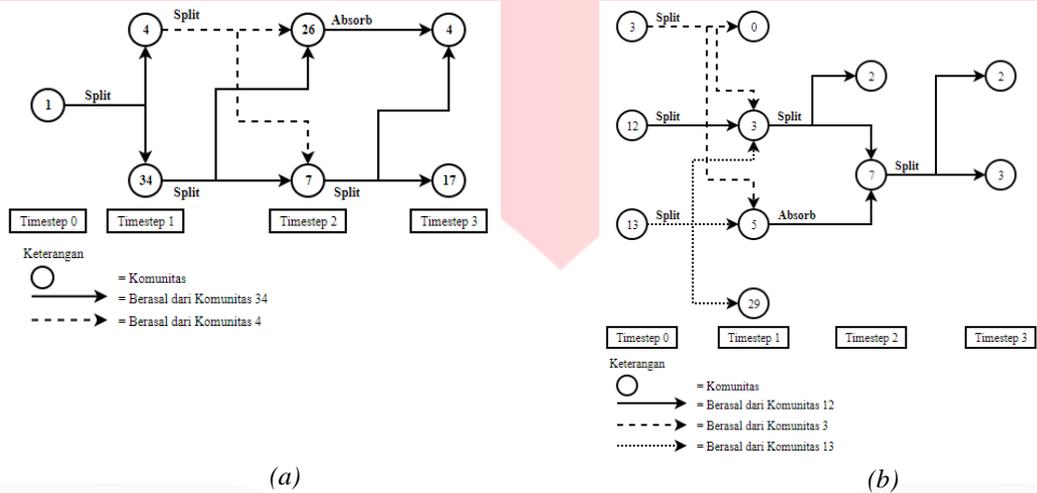
Tabel 2. NMI (a) 50 simpul, (b) 100 simpul

Nilai α, β, γ			50 simpul		
lifetime 0					
α	β	γ	ID Komunitas Timestep 0	ID Komunitas Timestep 1	Nilai NMI Komunitas timestep 0 terhadap timestep 1
0.33	0.33	0.33	0 - 49	0	0
			0 - 49	1	0
0.95	0.05	0	0 - 49	0	0
			0 - 49	1	0
lifetime 1					
α	β	γ	ID Komunitas Timestep 1	ID Komunitas Timestep 2	Nilai NMI Komunitas timestep 1 terhadap timestep 2
0.33	0.33	0.33	0 - 1	0	0
			0 - 1	1	0
0.95	0.05	0	0 - 1	0	0
			0 - 1	1	0
lifetime 2					
α	β	γ	ID Komunitas Timestep 2	ID Komunitas Timestep 3	Nilai NMI Komunitas timestep 2 terhadap timestep 3
0.33	0.33	0.33	0	0 - 49	0
			1	0 - 49	0
0.95	0.05	0	0	0 - 49	0
			1	0 - 49	0

Nilai α, β, γ			100 simpul		
lifetime 0					
α	β	γ	ID Komunitas Timestep 0	ID Komunitas Timestep 1	Nilai NMI Komunitas timestep 0 terhadap timestep 1
0.33	0.33	0.33	0	0	0
			0	1	0
0.98	0.02	0	0	0	0
			0	1	0
lifetime 1					
α	β	γ	ID Komunitas Timestep 1	ID Komunitas Timestep 2	Nilai NMI Komunitas timestep 1 terhadap timestep 2
0.33	0.33	0.33	0 - 1	0	0
			0 - 1	1	0
			0 - 1	2	0
0.98	0.02	0	0 - 1	0	0
			0 - 1	1	0
lifetime 2					
α	β	γ	ID Komunitas Timestep 2	ID Komunitas Timestep 3	Nilai NMI Komunitas timestep 2 terhadap timestep 3
0.33	0.33	0.33	0	0,1,2	0
			1	0,1,2	0
			2	0,1,2	0
0.98	0.02	0	0	0,1,2	0
			1	0,1,2	0
lifetime 2					
α	β	γ	ID Komunitas Timestep 2	ID Komunitas Timestep 3	Nilai NMI Komunitas timestep 2 terhadap timestep 3
0.33	0.33	0.33	0	0,1,2	0
			1	0,1,2	0
0.98	0.02	0	0	0,1,2	0
			1	0,1,2	0
lifetime 2					
α	β	γ	ID Komunitas Timestep 2	ID Komunitas Timestep 3	Nilai NMI Komunitas timestep 2 terhadap timestep 3
0.33	0.33	0.33	0	0,1,2	0
			1	0,1,2	0
0.98	0.02	0	0	0,1,2	0
			1	0,1,2	0

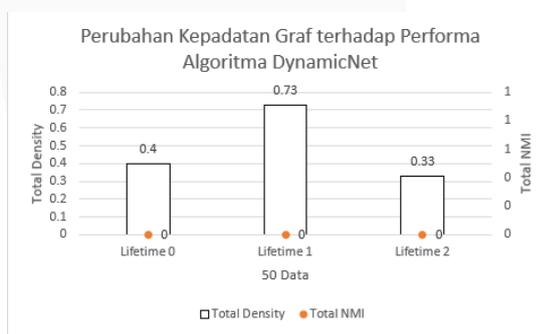
Gambar 7 dan Gambar 8 merepresentasikan jumlah komunitas pada setiap jenis evolusi yang dapat dilihat rinciannya pada Lampiran 3 dan Lampiran 4. Pada Gambar 8 dapat dilihat jika jenis evolusi yang terjadi hanya berupa *disappear* dan *appear* walaupun sudah ditingkatkan jumlah datanya dan dirubah nilai $\alpha\beta\gamma$ dimana hal ini berarti keempat data tidak memiliki kesamaan sama sekali, oleh karena itu perlu dilakukan perhitungan NMI untuk memastikan apakah benar keempat data tidak memiliki kesamaan sama sekali. Pada keempat tabel (Tabel

2, Lahiran 5A dan 5B) diberikan hasil perhitungan NMI dari setiap komunitas yang digunakan sebagai masukan algoritma DynamicNet. Jika dilihat dari tabel 2 diatas peningkatan jumlah data dari 50 menjadi 100 tidak ada perubahan pada nilai NMI sehingga evolusi yang terjadi tetap berupa *disappear* dan *appear*. Namun jika dilihat pada Lampiran 5A dan 5B terdapat perubahan pada nilai NMI walaupun evolusi yang terjadi masih berupa *disappear* dan *appear* seperti yang dilampirkan pada Lampiran 4A dan 4B, hal ini dikarenakan nilai threshold yang digunakan bernilai 0.5 sehingga nilai NMI dibawah 0.5 tidak akan masuk kedalam proses *match* yang mengakibatkan sistem menilai jika antar kedua data tidak memiliki komunitas yang sama. Hal ini menunjukkan jika data uji menggunakan 50 – 100 simpul memiliki kualitas yang tidak baik karena setiap data tidak memiliki kesamaan sama sekali

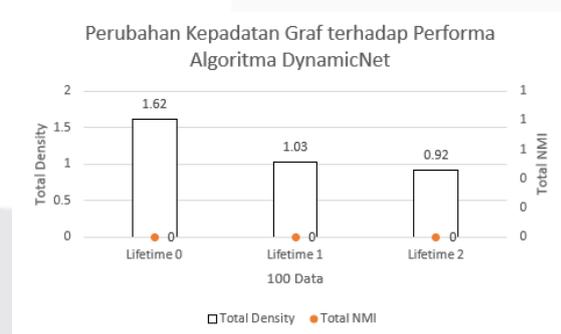


(a) (b)
Gambar 9 Proses evolusi pada (a) 1000 data dan (b) 1500 data

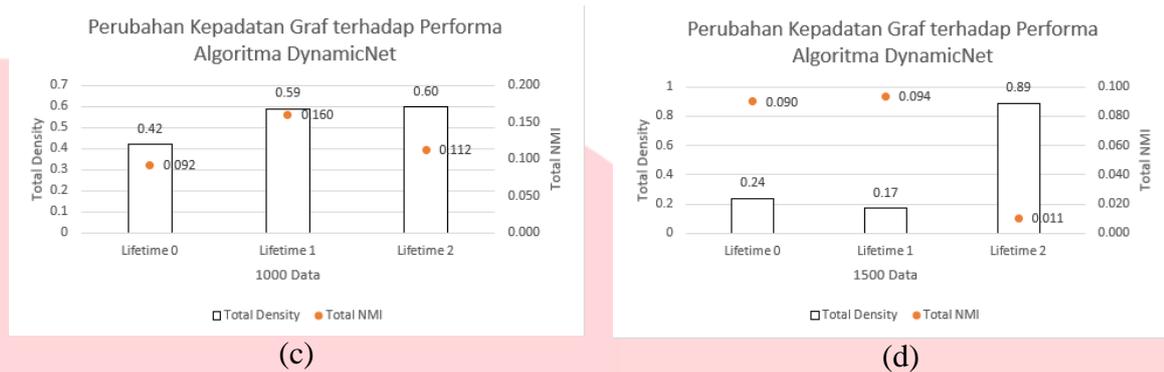
Pada point disebutkan jika hasil evolusi dari setiap data Timestep dan dari setiap jumlah data simpul hanya berupa *disappear* dan *appear* karena nilai threshold bernilai 0.5, oleh karena itu nilai threshold dikurangi menjadi 0.001. Pengambilan nilai 0.001 sendiri dikarenakan nilai NMI dari keseluruhan data paling kecil bernilai 0.005 oleh karena itu diambil nilai threshold 0.001 agar semua komunitas dapat dilihat proses evolusi aslinya. Setelah dilakukan perubahan nilai threshold evolusi yang terdeteksi menjadi beragam seperti yang dapat dilihat pada Gambar 9 yang merepresentasikan proses evolusi dari komunitas pada 1000 dan 1500 data simpul. Pada Gambar (a) dapat dilihat jika komunitas 1 dari Timestep 0 bisa terus *survive* dengan proses evolusi *split* hingga data Timestep 3, hal ini menunjukkan jika pengguna Twitter pada komunitas 1 dalam rentan waktu pengambilan data selalu melakukan interaksi. Pada Gambar (b) juga dapat dilihat jika komunitas 3,12 dan 13 dari data Timestep 0 sebagian besar dapat *survive* terus hingga akhir dengan proses *split* dan *absorb*. Dari kedua Gambar diatas dapat dilihat bahwa selain jumlah data memiliki peran dalam menambah banyaknya komunitas yang mengalami proses evolusi, threshold juga berperan untuk batas minimal NMI dari suatu komunitas terhadap komunitas lain agar dapat dikatakan berevolusi.



(a)

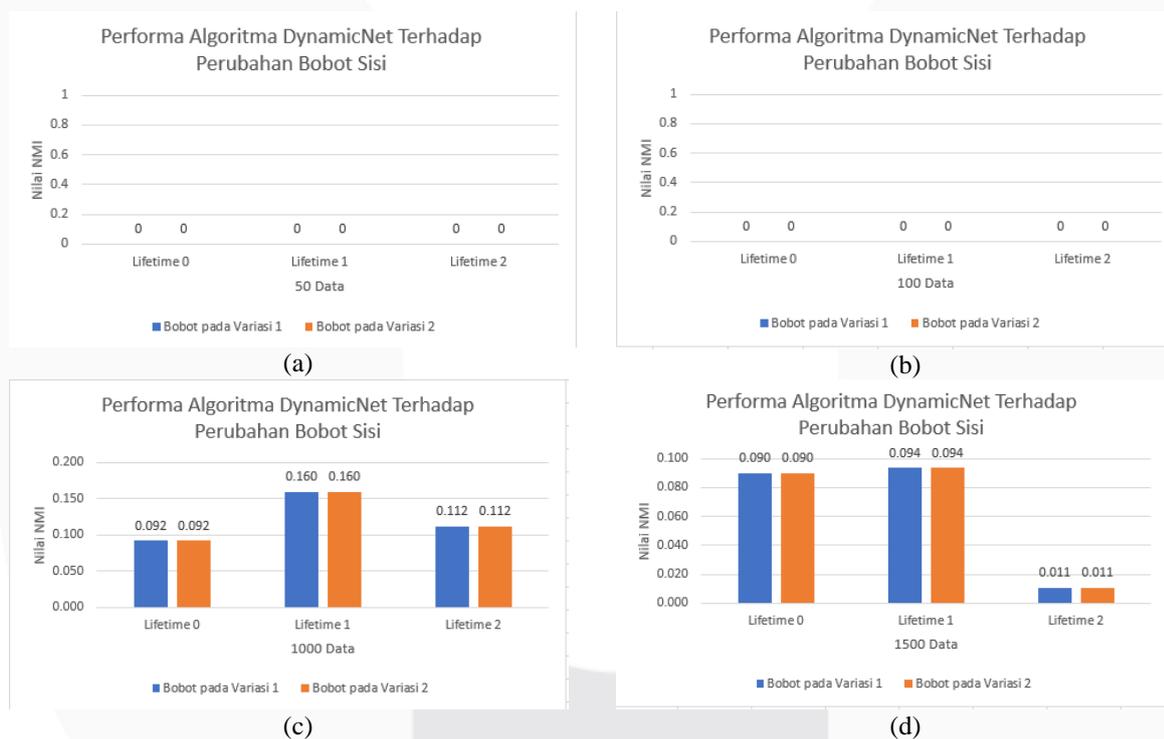


(b)



Gambar 10 Perubahan Kepadatan Graf Terhadap Performa DynamicNet (a) 50 Data, (b) 100 Data, (c) 1000 Data, (d) 1500 Data

Pada Gambar 1 diatas menunjukkan perubahan kepadatan graf (*Density*) terhadap performa algoritma DynamicNet yang dinilai menggunakan NMI. Pada Gambar 1(a) dan 1(b) dapat dilihat jika nilai NMI selalu bernilai 0 walaupun graf tersebut nilai kepadatannya naik ataupun turun, hal ini terjadi karena setiap data yang digunakan pada Lifetime 0, 1 dan 2 tidak memiliki kesamaan simpul sama sekali baik pada 50 Data ataupun 100 Data. Hasil yang berbeda dapat dilihat pada Gambar 1(c) dan 1(d) yang terjadi perubahan nilai total NMI yang berarti pada 1000 dan 1500 Data, data yang digunakan memiliki terdapat kesamaan simpul sehingga dapat menaikkan nilai NMI. Jika dilihat dari setiap Lifetime, perubahan nilai NMI cenderung mengikuti perubahan nilai Density, dapat dilihat untuk Lifetime 0 dari 50 hingga 1500 Data, nilai NMI dari Lifetime 0 selalu mengikuti perubahan nilai Density. Hal ini berarti jika kepadatan graf mempengaruhi performa dari algoritma DynamicNet sehingga untuk mendapatkan performa yang baik, harus digunakan graf yang memiliki kepadatan tinggi.



Gambar 11 Performa Algoritma DynamicNet Terhadap Perubahan Bobot Sisi

Gambar 11 di atas adalah perbandingan performa dari algoritma DynamicNet apabila bobot sisi yang digunakan diubah. Perubahan bobot sisi ini sendiri dilakukan menggunakan bobot yang dihasilkan oleh variasi nilai $\alpha\beta\gamma$ yang sudah dihitung terlebih dahulu pada proses perhitungan *similarity*. Dari hasil pengujian tersebut, dapat diketahui bahwa bobot sisi tidak mempengaruhi performa dari algoritma DynamicNet karena walaupun menggunakan bobot sisi yang berbeda, hasil yang didapatkan tetap sama seperti sebelum diubah.

Dari kedua hasil pengujian di atas didapatkan kesimpulan bahwa komunitas 3 pada Timestep 0 dengan jumlah data 1500 simpul dengan nilai threshold 0.001 menjadi puncak performansi dari kedua algoritma, karena kualitas komunitas (*modularity*) yang tinggi yaitu 0.968 dan juga komunitas tersebut *survive* hingga akhir pengujian. Hal

ini terjadi karena dengan jumlah simpul yang lebih banyak dibandingkan dengan data yang lain, sehingga memungkinkan adanya jumlah sisi yang padat didalam satu komunitas, dibuktikan oleh komunitas 3 itu sendiri. Komunitas 3 dapat survive hingga akhir pengujian karena memiliki nilai NMI yang lebih tinggi dibandingkan dengan komunitas lain, walaupun nilai NMInya masih dibawah 0.1.

5. Kesimpulan dan Saran

5.1. Kesimpulan

Dari hasil pengujian dan analisis kesimpulan yang didapat adalah:

1. Nilai (α, β, γ) tidak mempengaruhi hasil dari komunitas yang terbentuk. Nilai (α, β, γ) hanya mempengaruhi bobot dari setiap sisi (nilai *similarity*) yang akan digunakan.
2. Nilai modularity dari komunitas yang terbentuk dipengaruhi oleh jumlah relasi antar anggota komunitas yang mana semakin padat relasi di dalam satu komunitas maka nilai modularity akan semakin tinggi.
3. Performa terbaik algoritma DynamicNet terjadi saat menggunakan 1000 – 1500 Data yang memiliki nilai kepadatan 0.17 – 0.89 terdapat perubahan nilai NMI dimana puncaknya bernilai 0.112. Dari data tersebut menunjukkan untuk perubahan kepadatan graf 0.17 – 0.89 performa DynamicNet semakin membaik.
4. Bobot sisi pada penelitian ini tidak mempengaruhi performa dari algoritma DynamicNet karena menghasilkan performa yang sama setelah dilakukan perubahan bobot sisi.
5. Jumlah data, jumlah sisi, threshold berpengaruh dalam perubahan hasil dari algoritma DynamicNet karena dengan penambahan jumlah data hasil komunitas yang berevolusi menjadi lebih banyak, sehingga nilai NMI menjadi lebih baik.

5.2. Saran

Adapun saran untuk kedepannya adalah sebagai berikut:

1. Dalam implementasi algoritma DynamicNet, sebaiknya menggunakan jumlah data simpul maupun sisi yang lebih banyak, sehingga dapat menaikkan nilai NMI dan juga mendapatkan hasil evolusi yang lebih beragam.
2. Nilai threshold yang digunakan sebaiknya lebih rendah dibandingkan rata2 nilai NMI yang ada untuk menghindari tidak terdeteksinya evolusi yang terjadi.
3. Data komunitas yang digunakan sebaiknya memiliki nilai modularity yang baik, sehingga evolusi yang terjadi tidak terbatas pada *disappear* dan *appear* saja,

Daftar Pustaka

- [1] Cuzzocrea, Alfredo; Folini, Francesco; Pizzuti, Clara, "DynamicNet: An Effective and Efficient Algorithm for Supporting Community Evolution Detection in Time-Evolving Information Network," 2013.
- [2] Tang, Lei; Liu, Huan, "Graph Mining Applications to Social Network Analysis," in *Managing and Mining Graph Data*, Springer Science + Business Media, 2010.
- [3] Twitter, "About," [Online]. Available: www.about.twitter.com/company. [Accessed 5 5 2017].
- [4] Blondel, Vincent; Guillaume, Jean-Loup; Lambiotte, Renaud; Lefebvre, Etienne, "Fast Unfolding of Communities in Large Network," *Journal of Statistical Mechanics: Theory and Experiment*, 2008.
- [5] Li, Xiaoming; Wu, Bin; Guo, Xian; Zeng, Xuelin; Shi, Chuan, "Dynamic Community Detection Algorithm Based on Incremental Identification," *Advances in Social Networks Analysis and Mining (ASONAM) 2016 IEEE/ACM International Conference*, 2016.
- [6] Soo Kim, Min; Han, Jiawei, "A particle-and-density based evolutionary clustering method for dynamic networks," *Proceedings of the VLDB Endowment*, vol. II, Agustus 2009.
- [7] Prajapati, Juhi; Shah, S.M, "Improved Community Detection Algorithm based on Distance Centrality in Social Network," vol. II, 2016.
- [8] Takkafoli, Mansoureh; Rabbany, Reihaneh; Zaiane, Omar.R, "Community Evolution Prediction in Dynamic Social Networks," 2014.
- [9] Yang, Zhang, et. al., "Community Discovery in Twitter Based on User Interest," *Binary Information Press*, 2012.
- [10] Pan, Long, "Effective and Efficient Methodologies for Social Network Analysis," *State University*, 2012.
- [11] Zhang, Pan, "Evaluating accuracy of community detection using the relative normalized mutual information," *Journal of Statistical Mechanics: Theory and Experiment*, no. 11, 2015.
- [12] Danon, Leon; Díaz-Guilera, Albert; Duch, Jordi; Arenas, Alex, "Comparing community structure identification," *Journal of Statistical Mechanics: Theory and Experiment*, no. 09, 2005.
- [13] Ahuja, Mini & Singh, Jatinder, "FUTURE PROSPECTS IN COMMUNITY DETECTION," *International Journal of Computer Science Engineering and Information Technology Research (IJCSEITR)*, vol. 4, pp. 37-48, 2014.
- [14] Twitter, "Support," [Online]. Available: <https://support.twitter.com/groups/34-apps-sms-and-mobile/topics/153-twitter-via-sms/articles/14014-twitter-via-sms-faq#>. [Accessed 12 Oktober 2017].
- [15] Twitter, "Blog," [Online]. Available: https://blog.twitter.com/official/en_us/a/2012/twitter-turns-six.html. [Accessed 12 Oktober 2017].