Implementasi dan Analisis *Betweenness Centrality* Berbasis Konten Menggunakan Algoritma Geisberger

Joshua Tanuraharja[1], Warih Maharani[2], Alfian Akbar Gozali[3]

[1,2,3]Teknik Informatika, School of Computing, Universitas Telkom
Jalan Telekomunikasi No. 1, Dayeuhkolot, Bandung 40257
[1]joshua.tanuraharja@hotmail.com, [2] warihmaharani@gmail.com, [3] alfian.akbar.gozali@gmail.com

Abstrak

Jejaring sosial marak digunakan oleh Usaha Kecil Menengah untuk memasarkan produk dan jasa, guna mencari *user* yang berpotensi menjadi *buzzer* dibutuhkan pengetahuan tentang *Social Network Analysis* (SNA). SNA digunakan untuk menganalisis interaksi dalam suatu kelompok jaringan sosial. Contoh pengembangan SNA adalah *Content Based Social Network Analysis* (CBSNA) yang dapat digunakan untuk menentukan *rangking user* berpengaruh berdasarkan relasi kesamaan konten. Salah satu metode penghitungan *centrality* adalah metode *Linear Scaling* yang dikembangkan oleh Geisberger, dimana dalam menghitung *betweenness centrality* cukup menggunakan beberapa *node* sebagai sumber.

Pada penelitian tugas akhir ini metode yang digunakan adalah *Linear Scaling* yang dipadukan dengan *Vector Space Model*, pertama bertujuan untuk menghitung *betweenness centrality* berbasis konten pada studi kasus media sosial Twitter dan yang kedua untuk menganalisis parameter yang berpengaruh pada metode *Linear Scaling* dalam penghitungan nilai *betweenness centrality*. Hasil pengujian menunjukkan bahwa *user* dengan nilai *similarity* tinggi memiliki isi konten *Quote Retweet* selain itu metode *Linear Scaling* dipengaruhi oleh nilai pivot (k) dan jumlah *edge* suatu graf. *Linear Scaling* dapat digunakan untuk menghitung *betweenness centrality* guna menentukan *ranking user* yang berpengaruh berdasar suatu kata kunci tertentu.

Kata Kunci : Usaha Kecil Menengah, *Content Based Social Network Analysis*, *Betweeneess Centrality*, Algoritma Geisberger, *Linear Scaling*, *Vector Space Model.*

1. Pendahuluan

Pelaku bisnis mulai memanfaatkan sosial media sebagai bagian dari stategi pemasaran. Sosial media membantu penyebaran informasi dan edukasi produk kepada masyarat yang tidak dapat dijangkau oleh pelaku bisnis secara fisik, misal masyarakat dikota lain. Kemudahan dan rendahnya biaya promosi melalui sosial media seperti Twitter menjadi kesempatan untuk menjaring konsumen potensial yang belum terjamah oleh pelaku bisnis.Pelaku bisnis perlu menemukan orang yang tepat didalam jejaring sosial Twitter untuk menyebarkan informasi produknya. Ilmu yang mempelajari hal ini adalah *Social Network Analysis* (SNA). SNA didefinisikan sebagai proses pemetaan dan pengukuran relasi antara orang ke orang[16]. Pemetaan ini untuk memahami pola interaksi yang kompleks [2].

Content Based SNA (CBSNA) muncul sebagai pendekatan baru dalam *Social Network Analysis*

(SNA). Relasi pertemanan antar *user* bukan yang diutamakan melainkan kesamaan *interest* terhadap suatu *kata kunci* yang dipertimbangkan [11]. Sekumpulan *tweet* yang mengandung *query* tertentu dihitung *similarity*-nya terhadap *tweet* lain menggunakan *Vector Space Model* (VSM) sehingga terbentuk graf tidak berarah dan berbobot (*undirected-weighted graph*) dimana *node* adalah representasi *user* pemilik *tweet* tersebut.

Masalah tersebut dapat diatasi dengan metode *Linear Scaling* dari algoritma Geisberger [12][15], dimana dalam memperkirakan nilai *betweenness centrality* sebuah *node* tidak menggunakan seluruh jalur *shortest paths* namun menggunakan pendekatan berupa pengambilan sampel beberapa jalur *shortest paths* [12][13]. Oleh karena itu pada tugas akhir ini menggunakan metode *linear scaling* untuk mengetahui *ranking user* yang berpengaruh pada jejaring Twitter.

2. Landasan Teori

2.1 Social Network

Social network adalah struktur sosial yang terdiri dari aktor dan relasi antar aktor. Setiap aktor dilambangkan dengan sebuah "*node*", dimana setiap *node* terkoneksi dengan satu atau lebih *node* lain yang membentuk keterhubungan seperti, pertemanan, kekerabatan, kesamaan kepentingan, relasi kepercayaan, dll [5].

Streeter [2] Mendefinisikan *social network* sebagai *social unit* yang saling terhubung dan memiliki batas. Definisi ini menekankan pada tiga karakteristik utama pada *social network*, yaitu :

a. Jaringan memiliki batas
b. Keterhubungan pada *social network*
c. Terdiri dari *social unit*

2.1.1 Social Media Twitter

Twitter adalah sebuah *social network* dimana anggotanya saling berbagi informasi aktivitas yang dilakukan. M*icroblogging* Twitter hanya mampu mengantar pesan pendek, dengan panjang maksimal 140 karakter untuk setiap pesan [6].

Twitter merupakan jejaring sosial besar yang fokus pada kecepatan komunikasi. Kecepatan dan kemudahaan dalam hal publikasi pesan membuat Twitter menjadi media komunikasi yang penting. Kumar [14] mengatakan bahwa terdapat sekitar 140 juta pengguna aktif yang membuat lebih dari 400 juta pesan setiap sehari.

Twitter dapat digunakan oleh pengguna untuk mempublikasikan pesan ("*tweeting*") dengan sangat cepat dan mudah. Pengguna dapat terhubung dengan pengguna lain melalui fitur '*follow*', sehingga pengguna dapat mengikuti *tweet* terbaru dari pengguna yang dia *follow*. Perlu diperhatikan bahwa mekanisme *follow* tidak mewajibkan pengguna lain untuk melakukan *follow* balik.

Istilah lain yang cukup populer diantaranya adalah RT atau *retweet*, dengan simbol '@' yang diikuti nama pengguna. *Retweet* merupakan sarana membalas *tweet* dengan menyertakan isi *tweet* sumber, sehingga pengguna yang menerima *retweet* bisa memahami konteks pesan yang diterima. Tak kalah populer adalah simbol '#' yang diikuti sebuah kata yang merepresentasikan sebuah *hastag*. Opsi ini penting untuk menandai konteks dari sebuah pesan Twitter, namun *hastag* bukanlah syarat untuk publikasi *tweet*.

Terdapat beberapa alasan kenapa pesan Twitter digunakan sebagai sumber penelitian ini [1] :

1. Frekuensi *posting* pesan yang sangat tinggi.
2. Pesan Twitter tidak terlalu panjang hanya 140 karakter, sehingga lebih deskriptif dan mudah dimengerti.
3. Twitter menyediakan *semi-structured meta-data* (kota, negara, jenis kelamin, umur).

2.2 Social Network Analysis

Social Network Analysys (SNA) adalah sekumpulan metode yang digunakan untuk menginvestigasi aspek relasi pada struktur sosial [3] [5]. Krebs [16] mendefinisikan SNA sebagai proses pemetaan dan pengukuran relasi antara orang ke orang. Pemetaan ini untuk memahami pola interaksi yang kompleks [2]. Otte dan Rousseau [9] menambahkan bahwa SNA dapat digunakan untuk pengambilan informasi yang implisit, seperti hubungan pertemanan.

SNA terfokus pada pola interaksi antar entitas. Dengan kata lain, SNA lebih memperhatikan hubungan antar aktor daripada aktor itu sendiri. Manfaat penggunaan SNA antara lain [8]:

a. Mengidentifikasi individu, kelompok, dan unit yang memiliki peran sentral.
b. Meningkatkan inovasi dan pemahaman
c. Memperbaiki strategi.
d. Meningkatkan efisiensi *channel* komunikasi formal.

2.3 *Centrality* Measurement

Scoot [5] mendefinisikan *centrality measurement* sebagai sebuah pengukuran yang menghasilkan perkiraan kasar dari *social power* sebuah *node* berdasarkan seberapa baik *node* tersebut terkoneksi didalam jaringan. Semakin besar nilai *social power* dari sebuah *node*, menandakan *node* tersebut memiliki andil yang cukup besar dalam pendistribusian informasi. Menurut Otte dan Rousseau [9] dan Scoot [5] terdapat tiga jenis metode *centrality* yang paling sering digunakan, yaitu : *Degree centrality*, *Closeness centrality* dan *Betweenness centrality*.

2.4 Beetweeness *Centrality*

Betweenness sama terkenalnya dengan *degree* dan *closeness*. Scoot [5] mendefinisikan bahwa *node* yang paling sering dilewati oleh *shortest paths* atau jalur terpendek memiliki nilai *betweenness centrality* yang lebih besar dari pada yang tidak. Penghitungan *Betweenness centrality* secara matematis dapat dilakukan dengan persamaan berikut [7]:

$$C_B(v) = \sum_{s \neq v} \sum_{v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}} \quad (2.1)$$

keterangan:

$\sigma_{st}(v)$ = jumlah *shortest paths* dari *node* s ke t yang melewati *node v*

σ_{st} = jumlah *shortest paths* dari *node* s ke t

2.5 Algoritma Geisberger

Geisberger [12] melihat adanya *overestimated node* pada *node-node* kurang penting yang berada didekat *pivot* jika menggunakan algoritma Brandes. Geisberger mengembangkan pendekatan *betweenness* yang lebih baik dari pendekatan Brandes dengan cara menerapkan *Liniear Scaling*. Metode *Liniear scaling* mudah diimplementasikan dengan menggunakan nilai *edge* asli sebagai panjang pada fungsi L. Implementasinya hanya merubah sedikit rumus dari skema Brandes.

$$\sigma_s(v) = \sum_{w \in succ(v)} \frac{\mu(s,v)}{\mu(s,w)} \cdot \frac{\sigma_{sv}}{\sigma_{sw}} (1 + \sigma_s(w)) \quad (2.3)$$

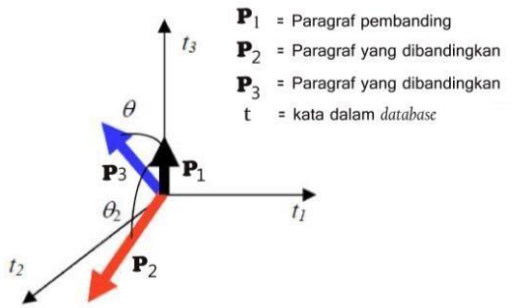
Keterangan :

$\mu(s,w)$= jarak terpendek dari s ke w

$\mu(s,v)$ = jarak terpendek dari s ke v

2.6 Vector Space Model

Vector Space model sering digunakan untuk merepresentasikan sebuah dokumen dalam ruang vektor [10]. VSM digunakan untuk mengukur kemiripan antara dua dokumen yang berbeda. Nilai kosinus digunakan untuk menghitung tingkat kesamaan antar dua dokumen yang dibandingkan.



Gambar 2.1 Vector Space Model

D_i merepresentasikan setiap dokumen dan d_{it} merepresentasikan setiap istilah didalam dokumen.

$$D_i = (d_{i1}, d_{i2}, d_{i3}, \dots, d_{it}) \quad (2.4)$$

D_1 adalah vektor dari dokumen pembanding D_2 adalah vetor dari dokumen yang dibandingkan. Simbol ɵ merepresentasikan sudut yang terbentuk oleh kedua dokumen vektor.

$$D_1 . D_2 = |D_1||D_2| \cos(\theta) \quad (2.5)$$

Sedangkan $|D_1|$ dan $|D_2|$ adalah panjang vektor.

$$|D_1| = \sqrt{\sum_{i=1}^{n} D_1 i^2} \quad (2.6)$$

$$|D_2| = \sqrt{\sum_{i=2}^{n} D_2 i^2} \quad (2.7)$$

Selanjutnya tingkat kesamaan dari dua dokumen dihitung dengan rumus kosinus. Jika menghasilkan nilai 1, artinya kedua komen sama persis. Jika menghasilkan nilai 0, artinya tidak ada kesamaan sama sekali.

$$Sim(D_1, D_2) = Cos\ \theta = \frac{D_1 . D_2}{|D_1||D_2|} \quad (2.8)$$

VSM digunakan untuk membantu mencari nilai kesamaan suatu dokumen dengan dokumen lainnya dengan memanfaatkan matriks TF-IDF.

2.7 UKM

Usaha Kecil Menengah (UKM) adalah kegiatan ekonomi rakyat yang berskala kecil dengan bidang usaha yang secara mayoritas merupakan kegiatan usaha kecil dan perlu dilindungi untuk mencegah dari persaingan usaha yang tidak sehat.[4]

Adapun kriteria dari UKM (Usaha Kecil Menengah) [4] antara lain:

1. Memiliki kekayaan bersih paling banyak Rp. 200.000.000,- (Dua Ratus Juta Rupiah) tidak termasuk tanah dan bangunan tempat usaha.
2. Memiliki hasil penjualan tahunan paling banyak Rp. 1.000.000.000,- (Satu Miliar Rupiah).
3. Milik Warga Negara Indonesia.
4. Berdiri sendiri, bukan merupakan anak perusahaan atau cabang perusahaan yang tidak dimiliki, dikuasai, atau berafiliasi baik langsung maupun tidak langsung dengan Usaha Menengah atau Usaha Besar.
5. Berbentuk usaha orang perorangan, badan usaha yang tidak berbadan hukum, atau badan usaha yang berbadan hukum, termasuk koperasi.

3. Perancangan Sistem

3.1 Gambaran Umum Sistem

Secara umum proses yang dilakukan adalah pengumpulan *dataset* sesuai dengan kata kunci dari UKM @jkt66official menggunakan fitur *crawling*. Ketika *dataset* sudah tersedia proses selanjutnya adalah *preprocessing* dengan memilih sejumlah N

node yang memiliki jumlah *tweet* terbanyak, kemudian dilanjutkan dengan proses penghitungan nilai *similarity* antar *node* menggunakan VSM. Ketika nilai similarity antar node sudah diperoleh maka dapat dilakukan penghitungan nilai *betweenness centrality* menggunakan metode *linear scaling*.

3.2 Contoh Implementasi VSM dengan TF-IDF

Contoh implementasi sederhana dari TF-IDF adalah sebagai berikut :

Kata kunci (kk) = saya makan nasi padang
Dokumen 1 (D1) = Saya makan nasi padang
Dokumen 2 (D2) = Nasi goreng enak
Dokumen 3 (D3) = Makan nasi ayam
Dokumen 4 (D4) = Masak Nasi dulu
Dokumen 5 (D5) = Makan tahu goreng

Tabel 3.1 Implementasi VSM dengan TF-IDF

Token	TF						df	D/df	IDF=	Bobot					
	KK	D1	D2	D3	D4	D5			1+Log(D	KK	D1	D2	D3	D4	D5
saya	1	1	0	0	0	0	1	5	1.699	1.699	1.699	0.000	0.000	0.000	0.000
makan	1	1	0	1	0	1	3	1.66667	1.222	1.222	1.222	0.000	1.222	0.000	1.222
nasi	1	1	1	1	1	0	4	1.25	1.097	1.097	1.097	1.097	1.097	1.097	0.000
padang	1	1	0	0	0	0	1	5	1.699	1.699	1.699	0.000	0.000	0.000	0.000
goreng	0	0	1	0	0	1	2	2.5	1.398	0.000	0.000	1.398	0.000	0.000	1.398
enak	0	0	1	0	0	0	1	5	1.699	0.000	0.000	1.699	0.000	0.000	0.000
ayam	0	0	0	1	0	1	2	2.5	1.398	0.000	0.000	0.000	1.398	0.000	1.398
masak	0	0	0	0	1	0	1	5	1.699	0.000	0.000	0.000	0.000	1.699	0.000
dulu	0	0	0	0	1	0	1	5	1.699	0.000	0.000	0.000	0.000	1.699	0.000

Pada Tabel 3.2 dilakukan penghitungan *term*(tf), *document frequency*(df) serta *inverse document frequency*(idf) guna menghitung *similarity* antar dokumen dengan menggunakan metode TF-IDF.

Tabel 3.2 Implementasi VSM dengan Cosine *Similarity*

Token	KK	D1	D2	D3	D4	D5	KK*D1	KK*D2	KK*D3	KK*D4	KK*D5
saya	1.699	1.699	0	0	0	0	2.886	0	0	0	0
makan	1.222	1.222	0	1.222	0	1.222	1.493	0	1.49291	0	1.493
nasi	1.097	1.097	1.097	1.097	1.097	0	1.203	1.20321	1.20321	1.203212	0
padang	1.699	1.699	0	0	0	0	2.886	0	0	0	0
goreng	0	0	1.398	0	0	1.398	0	0	0	0	0
enak	0	0	1.699	0	0	0	0	0	0	0	0
ayam	0	0	0	1.398	0	1.398	0	0	0	0	0
masak	0	0	0	0	1.699	0	0	0	0	0	0
dulu	0	0	0	0	1.699	0	0	0	0	0	0
	Sqrt(KK	Sqrt(Di)					Sum(KK.Di)				
	2.91	2.91	2.458	2.156	2.641	2.324	8.469	1.20321	2.69613	1.203212	1.493

Pada Tabel 3.3 dilakukan penghitungan lanjutan untuk mencari nilai *similarity* antara katakunci dengan dokumen menggunakan metode *Cosine Similarity*. *Cosine Similarity* digunakan untuk membantu TF-IDF agar *similarity* lebih presisi.

cos(d1,kk)=	sum(kk.d1)/((sqrt(kk)*sqrt(d1))= 1
cos(d2,kk)=	sum(kk.d2)/((sqrt(kk)*sqrt(d2))= 0.168
cos(d3,kk)=	sum(kk.d3)/((sqrt(kk)*sqrt(d3))= 0.43
cos(d4,kk)=	sum(kk.d4)/((sqrt(kk)*sqrt(d4))= 0.157
cos(d5,kk)=	sum(kk.d5)/((sqrt(kk)*sqrt(d5))= 0.221

Tabel 3.3 Hasil Cosine *Similarity*

Berdasarkan nilai *similarity* diatas, urutan dokumen yang akan ditampilkan adalah D1, D3, D5, D2, D4

3.3 Contoh Implementasi Metode Linier Scaling

Sebagai contoh diberikan empat pesan Twitter :

A = “Aku suka durian”
B = “Sedang musim durian”
C = “Musim hujan dan musim durian”
D = “Aku suka durian dan nanas”

Misal pesan Twitter tersebut sudah dihitung bobotnya dengan *Vector Space Model*, sehingga terbentuk graf dengan 4 *node* dan 6 *edge*.

Graf tersebut ditampilkan sebagai matriks ketetanggaan sebagai berikut:

Tabel 3.4 Matriks Ketetanggan

	A	B	C	D
A	0	0.3	0.1	0.7
B	0.3	0	0.6	0.5
C	0.1	0.6	0	0.2
D	0.7	0.5	0.2	0

Berdasarkan Tabel 3.2 tiap sel merepresentasikan bobot *similarity* antar *node*. Tahap selanjutnya adalah proses penghitungan *betweenness centrality* dengan menggunakan metode *Linear Scaling*.

1. n diinisialisasi dengan jumlah *node*, maka n=4.
2. k diinisialisasi dengan angka yang jauh lebih kecil dari n, misalkan k=2.

3. Inisialisasi c[v] dari semua *node* anggota V dengan nilai 0.
   c[A]=0
   c[B]=0
   c[C]=0
   c[D]=0
4. Untuk i = 1 dari k maka lakukan langkah 5.
5. Pilih satu dari semua *node* v anggota V sebagai *node start*, misalkan *node* A. Kemudian pilih satu dari semua *node* yang ada sebagai *node* destination, misalkan *node* B. Pemilihan *node* dilakukan dengan pendekatan distribusi *uniform*.
6. Tentukan *shortest paths* dengan algoritma Djikstra, sehingga didapat A-C-D.
7. Untuk setiap *node* v anggota V maka variable c_{local} [v] diinisialisasi dengan 0.

$c_{local}[A] = 0$

$c_{local}[B] = 0$

$c_{local}[C] = 0$

$c_{local}[D] = 0$

8. Untuk semua *node* w anggota V yang bisa dicapai melalui *node start* dan bukan *node start* maka lanjut ke langkah 9.
9. *Node* yang bisa dicapai dari *node* A untuk arah A-C-D adalah D lalu C. Urutan penghitungan dari *node* dengan jarak paling jauh, kemudian menurun. Sehingga penghitungannya sebagai berikut :

- Hitung *node* D
  c[D] = 0+0*0.3 =0
  v = C
  $c_{local}[C]$= 0+0+1/0.3
  = 3.3
- Hitung *node* C
  c[C] = 0+3.3*0.1 =0.33
  v = A
  $c_{local}[A]$= 0+3.3+1/0.1
  = 13.3

10. Selanjutnya adalah i=2, kita pilih *node* C sebagai *start* dan *node* B sebagai *destination*. Sehingga diperoleh *shortest path* C-A-B.

- Hitung *node* B
  c[B] = 0+0*0.4 =0
  v = A
  $c_{local}[A]$= 0 + 0 +1/0.4
  = 2.5
- Hitung *node* A
  c[A] = 0+2.5*0.1 =0.25
  v = C
  $c_{local}[C]$ = 0+0+1/0.1
  = 10

11. Langkah terakhir adalah mengalikan c[v] dengan $\frac{2n}{k}$ sehingga diperoleh *betweenness centrality* seperti berikut : c[A]=0.25 x 2 x 4 /2 = 1 c[B]=0 x 2 x 4 /2 = 0 c[C]=0.33 x 2 x 4 /2 = 1.32 c[D]=0 x 2 x 4 /2 = 0
12. Proses terakhir yaitu perangkingan berdasarkan nilai *centrality*, sehingga didapat hasil seperti berikut :

Tabel 3.5 Hasil Perangkingan

Rank	Node	Centrality
1	C	1.32
2	A	1
3	B	0
4	D	0

13. Metode *Linear Scaling* ini menggunakan pendekatan dalam mencari nilai *centrality* sehingga tidak semua *node* akan dicoba sebagai *start* dan *destination*. Namun jumlah pendekatan ditentukan oleh parameter *pivot* (k).

4. Pengujian dan Analisis

Adapun penelitian ini bertujuan untuk Mengetahui 10 *user* berpengaruh beserta nilai *betweenness centrality* setiap *user* pada masing-masing *kata kunci* serta mengetahui pengaruh variable pivot (k) dan jumlah *edge* dalam penghitungan nilai *betweenness centrality*. Dataset yang digunakan adalah sejumlah *tweet* dari kata kunci tertentu yang berhubunganj dengan produk UKM @jkt66official. Dataset diambil dalam rentang bulan November hingga Desember 2014 dengan pusat kota Bandung, Indonesia. Kata kunci yang dipilih mengacu pada produk populer yang dimiliki UKM @jkt66official.

Tabel 4.1 *Dataset* Twitter

Kata Kunci	**ID *kata kunci***	**Jumlah Tweet**	**Jumlah *User***
Indomie	102	145	129
Martabak	103	985	916
Selai	104	22	11
Brownies	105	196	127
Kue Cubit	106	525	473

4.4 Analisis Pengaruh Nilai k dalam penghitungan Nilai *Centrality*

Pada pengujian ini dilakukan pengujian dari k=1 hingga k=10, dilanjutkan k=20 hingga k=100 dengan kelipatan 10 tergantung pada jumlah *user* pada masing-masing kata kunci. Pengujian ini bertujuan untuk melihat pengaruh Nilai k terhadap hasil perangkingan dan menentukan nilai k dengan syarat k<<N. Nilai k harus sangat kecil daripada jumlah total *node*, maka dengan ini nilai k yang akan dipilih maksimal bernilai 10% dari total *node* dan menghasilkan perangkingan yang cukup stabil.

4.4.1 Analisis Pengaruh Nilai k Terhadap Hasil Perangkingan

Pada skema ini dilakukan pengujian untuk kata kunci "Indomie", "Martabak", "Selai", "Brownies", "Kue Cubit", "@jkt66office" dan "Greentea" guna mencari nilai k ideal untuk masing-masing kata kunci. Nilai k ideal dapat digunakan sebagai acuan untuk menghitung nilai *betweenness centrality*. Nilai k sesedikit mungkin namun menghasilkan perangkingan yang mendekati perangkingan dengan k maksimum. Dibawah ini ditampilkan 3 tabel hasil pengujian yang dapat mewakili secara keseluruhan, sedangkan untuk tabel hasil lainnya dapat dilihat dalam lampiran.

Tabel 4.2 Pengaruh Nilai k=1 Terhadap Hasil Perangkingan "Kue Cubit"

Rank	uji 1	uji 2	uji 3	uji 4	uji 5
1	29	167	42	95	42
2	95	42	3	42	100
3	42	99	191	36	3
4	2	21	2	117	2
5	200	148	29	3	143
6	135	95	117	43	81
7	36	2	81	2	82
8	23	56	17	191	6
9	82	69	82	167	35
10	48	17	34	183	117

Berdasarkan Tabel 4.2 dapat dilihat bahwa hasil perangkingan *centrality* tidak stabil jika menggunakan k=1 karena *path* yang dilewati hanya satu dan *random*, sehingga *paths* yang diambil bisa tidak memiliki *node* perantara atau *path* yang dilewati memiliki *node* perantara yang berbeda setiap pengujian. Bahkan *node* dengan peringkat pertama belum dapat ditentukan karena *node* yang berada pada peringkat pertama berbeda pada setiap pengujian.

Tabel 4.3 Pengaruh Nilai k=100 Terhadap Hasil Perangkingan "Kue Cubit"

Rank	uji 1	uji 2	uji 3	uji 4	uji 5
1	42	117	42	42	117
2	191	42	191	108	42
3	108	191	108	143	108
4	2	2	2	82	2
Rank	uji 1	uji 2	uji 3	uji 4	uji 5
5	62	143	62	2	143
6	82	82	3	62	3
7	94	62	94	94	94
8	35	95	95	95	95
9	95	34	82	34	92
10	3	179	81	6	6

Berdasarkan Tabel 4.3 dapat dilihat bahwa hasil perangkingan dengan k=100 lebih stabil daripada hasil perangkingan pada Tabel 4.2. Hal ini disebabkan karena k=100 bernilai 50% dari total *node* milik *kata kunci* "Kue Cubit" sehingga lebih banyak *node* yang digunakan sebagai sumber dalam menentukan *shortest paths* dan berimbas pada semakin stabilnya hasil perangkingan. *Paths* yang dilalui adalah acak dan tak berulang maka dimungkinkan terjadi perbedaan posisi *node* pada setiap pengujian.

Tabel 4.4 Pengaruh Nilai k=200 Terhadap Hasil Perangkingan "Kue Cubit"

Rank	uji 1	uji 2	uji 3	uji 4	uji 5
1	117	117	117	117	117
2	42	42	42	42	42
3	191	191	191	191	191
4	2	2	2	2	2
5	82	82	82	82	82
6	108	108	108	108	108
7	95	95	95	95	95
8	3	3	3	3	3
9	143	143	143	143	143
10	62	62	62	62	62

Berdasarkan Tabel 4.4 dapat dilihat bahwa *node* hasil perangkingan dari pengujian pertama hingga pengujian ke lima mirip satu sama lain. Hal ini terjadi karena nilai k adalah 200, dengan demikian semua *node* pasti pernah menjadi *source* dari *shortest paths* meskipun pemilihan *node* dilakukan secara acak. Hasil pengujian k=200 dapat dianggap sebagai hasil pengujian *betweenness* paling baik untuk kata kunci "Kue Cubit". Pengujian dengan k yang semakin besar akan menghasilkan hasil perangkingan yang semakin stabil, karena semakin besar k yang digunakan maka semakin besar pula jumlah *node* yang digunakan sebagai *node* sumber untuk menghitung *shortest paths*.

4.5 Analisis Pengaruh Jumlah *Edge* Terhadap Perangkingan

Pada skenario ini dilakukan pengujian sebanyak 20 kali dimana masing-masing pengujian dilakukan sebanyak 5 kali untuk kata kunci "Indomie", "Martabak", "Selai", "Brownies", "Kue Cubit", "@jkt66office" dan "Greentea" guna melihat dampak pemotongan *edge* terhadap hasil perangkingan. Pengujian menggunakan k=1 hingga k=10 dengan kelipatan 1(satu) serta k=10% hingga k=100% dengan kelipatan 10%.

Pengujian dilakukan dengan melakukan pemotongan *edge* sebesar 25%, 50% dan 75% pada masing- masing graf. Pemotongan *edge* dilakukan dengan cara menghilangkan sejumlah *edge* yang memiliki nilai *similarity* rendah dengan menggunakan *threshold* tertentu agar dapat melakukan pemotongan *edge* dengan tepat. Hasil pemotongan *edge* akan menghasilkan graf baru yang dapat digunakan untuk melihat perubahan *rangking betweenness centrality*.

Guna melihat dampak pemotongan *edge* maka dibawah ini ditampilkan 4 tabel hasil uji dari kata kunci "Kue Cubit" yang dapat mewakili pengujian secara keseluruhan.

Tabel 4.5 Pengaruh *Edge* Terhadap Perangkingan "Kue Cubit" dengan k=1

POTONG 0%						POTONG 25%						POTONG 50%						POTONG 75%					
k=1						k=1						k=1						k=1					
Rank	uji1	uji2	uji3	uji4	uji5	Rank	uji1	uji2	uji3	uji4	uji5	Rank	uji1	uji2	uji3	uji4	uji5	Rank	uji1	uji2	uji3	uji4	uji5
1	29	167	42	95	42	1	42	42	3	95	95	1	42	126	3	117	42	1	42	95	82	29	95
2	95	42	3	42	100	2	126	126	155	42	42	2	167	23	35	42	126	2	95	2	2	95	2
3	42	99	191	36	3	3	95	95	54	3	3	3	2	42	6	27	95	3	126	101	42	42	42
4	2	21	2	117	2	4	100	3	173	2	2	4	3	35	52	2	3	4	6	167	185	2	200
5	200	148	29	3	143	5	35	17	42	167	167	5	193	95	42	81	17	5	144	82	71	200	167
6	135	95	117	43	81	6	144	100	27	193	193	6	191	118	21	191	100	6	3	42	6	135	185
7	36	2	81	2	82	7	185	6	192	185	185	7	46	191	200	143	6	7	185	17	33	36	36
8	23	56	17	191	6	8	34	21	189	22	22	8	82	10	105	94	21	8	153	34	17	52	17
9	82	69	82	167	35	9	117	89	2	6	6	9	23	3	154	82	89	9	191	6	95	48	28
10	48	17	34	183	117	10	27	191	114	191	191	10	113	82	82	34	191	10	100	22	189	23	33

Berdasarkan Tabel 4.5 dapat dilihat bahwa dengan menggunakan k=1 pada *kata kunci* "Kue Cubit" didapatkan hasil perangkingan yang tidak stabil. *Node* yang masuk dalam peringkat 10 besar sangat bervariasi. Variasi ini mengakibatkan pola peringkat menjadi tidak stabil dan tidak dapat digunakan untuk mendefinisikan *node* pada peringkat pertama. Ketidakstabilan peringkat ini terjadi pada seluruh graf yang diujikan sehingga pengaruh perubahan graf tidak dapat diamati.

Tabel 4.6 Pengaruh *Edge* Terhadap Perangkingan "Kue Cubit" dengan k=100

POTONG 0%						POTONG 25%						POTONG 50%						POTONG 75%					
k=50%	100					k=50%	100					k=50%	100					k=50%	100				
Rank	uji1	uji2	uji3	uji4	uji5	Rank	uji1	uji2	uji3	uji4	uji5	Rank	uji1	uji2	uji3	uji4	uji5	Rank	uji1	uji2	uji3	uji4	uji5
1	42	117	117	42	117	1	42	42	42	42	42	1	117	42	117	117	42	1	29	42	117	117	117
2	2	42	42	108	42	2	2	143	108	191	108	2	42	191	42	42	2	2	42	108	29	42	42
3	3	82	191	191	143	3	191	62	82	3	2	3	191	2	108	191	143	3	191	143	42	191	108
4	62	2	82	62	62	4	82	117	117	143	143	4	108	143	191	108	62	4	82	191	191	108	143
5	117	143	108	82	94	5	143	34	95	82	3	5	2	62	2	2	191	5	192	2	3	143	3
6	34	62	143	95	34	6	94	94	34	117	62	6	143	117	62	143	82	6	117	95	62	82	192
7	94	3	95	2	95	7	95	2	2	95	95	7	62	95	94	3	95	7	2	94	192	3	94
8	95	95	34	117	82	8	35	95	92	35	35	8	82	34	95	94	179	8	94	81	95	62	2
9	35	34	81	81	2	9	179	35	179	2	81	9	95	179	179	95	81	9	95	35	82	94	81
10	179	179	92	3	35	10	144	82	6	92	82	10	6	144	6	92	144	10	97	92	81	34	179

Berdasarkan Tabel 4.6 dapat dilihat bahwa hasil perangkingan "Kue Cubit" semakin stabil ditandai dengan munculnya *node* 42 dan 117 yang mendominasi peringkat pertama dari total pengujian. Jika dibandingkan dengan pengujian pada Tabel 4.5 terlihat jelas bahwa posisi peringkat pertama dihuni oleh dua *node* spesifik meskipun ada satu pengujian yang memiliki peringkat pertama *node* 29. Susunan pengisi peringkat 10 besar pun mulai stabil dengan mulai seringnya suatu *node* muncul pada hasil uji, contohnya adalah *node* 42 yang tidak pernah absen masuk peringkat 10 besar dari 20 pengujian. Perubahan jumlah *edge* pada 4 model graf yang diujikan belum terlihat memberi dampak signifikan pada perubahan peringkat. Pengaruh jumlah k masih lebih kuat dibandingkan pengaruh pemotongan *edge* pada graf

Tabel 4.7 Pengaruh *Edge* Terhadap Perangkingan "Kue Cubit" dengan k= 160

POTONG 0%						POTONG 25%						POTONG 50%						POTONG 50%					
k=80%	160					k=80%	160					k=80%	160					k=80%	160				
Rank	uji1	uji2	uji3	uji4	uji5	Rank	uji1	uji2	uji3	uji4	uji5	Rank	uji1	uji2	uji3	uji4	uji5	Rank	uji1	uji2	uji3	uji4	uji5
1	42	117	117	42	42	1	117	117	117	42	42	1	117	117	42	42	42	1	42	117	42	42	42
2	191	42	42	2	191	2	42	42	42	191	191	2	42	42	191	191	108	2	29	42	29	117	29
3	2	191	2	191	2	3	191	191	191	2	2	3	191	191	2	82	191	3	143	29	191	29	108
4	3	2	82	143	82	4	2	3	2	82	82	4	2	2	108	108	143	4	191	108	108	191	143
5	95	82	108	3	143	5	82	143	108	108	108	5	108	108	143	143	3	5	2	2	143	108	2
6	62	143	191	95	3	6	108	82	3	3	143	6	82	82	3	3	2	6	95	82	2	2	191
7	117	62	143	62	62	7	143	62	143	143	95	7	143	143	95	95	82	7	62	191	95	82	82
8	82	95	3	117	95	8	3	95	95	62	62	8	3	95	62	2	95	8	82	95	62	95	3
9	34	94	95	82	117	9	95	34	94	95	117	9	95	3	117	62	62	9	192	62	192	62	62
10	94	34	62	94	34	10	62	2	81	94	34	10	62	94	94	117	117	10	117	94	34	34	192

Berdasarkan Tabel 4.7 dapat dilihat bahwa *node* pengisi peringkat 10 besar semakin stabil meskipun urutan *node* berfluktuasi. Sama seperti Tabel 4.6, *node* 42 tidak pernah absen pada setiap hasil pengujian dari 4 model graf yang diujikan, posisi *node* 42 pun semakin stabil pada posisi peringkat satu dan peringkat dua. Tidak hanya *node* 42, *node* 191 juga tidak pernah absen dari 20 hasil uji diatas. Ditinjau dari perubahan hasil *ranking* pada 4 model graf yang digunakan dapat dilihat bahwa hasil perangkingan tidak berbeda jauh.

Tabel 4.8 Pengaruh *Edge* Terhadap Perangkingan "Kue Cubit" dengan k= 200

POTONG 0%						POTONG 25%						POTONG 50%						POTONG 75%					
k=100%	200					k=100%	200					k=100%	200					k=100	200				
Rank	uji1	uji2	uji3	uji4	uji5	Rank	uji1	uji2	uji3	uji4	uji5	Rank	uji1	uji2	uji3	uji4	uji5	Rank	uji1	uji2	uji3	uji4	uji5
1	117	117	117	117	117	1	117	117	117	117	117	1	117	117	117	117	117	1	117	117	117	117	117
2	42	42	42	42	42	2	42	42	42	42	42	2	42	42	42	42	42	2	42	42	42	42	42
3	191	191	191	191	191	3	191	191	191	191	191	3	191	191	191	191	191	3	29	29	29	29	29
4	2	2	2	2	2	4	2	2	2	2	2	4	2	2	2	2	2	4	191	191	191	191	191
5	82	82	82	82	82	5	82	82	82	82	82	5	82	82	82	82	82	5	108	108	108	108	108
6	108	108	108	108	108	6	108	108	108	108	108	6	108	108	108	108	108	6	2	2	2	2	2
7	95	95	95	95	95	7	95	95	95	95	95	7	95	95	95	95	95	7	143	143	143	143	143
8	3	3	3	3	3	8	3	3	3	3	3	8	3	3	3	3	3	8	82	82	82	82	82
9	143	143	143	143	143	9	143	143	143	143	143	9	143	143	143	143	143	9	95	95	95	95	95
10	62	62	62	62	62	10	62	62	62	62	62	10	62	62	62	62	62	10	3	3	3	3	3

Berdasarkan Tabel 4.8 dapat dilihat bahwa hasil perangkingan berubah ketika graf mengalami pemotongan *edge* sebesar 75%. Hasil perangkingan dengan tanpa pemotongan *edge*, pemotongan *edge* sebesar 25% dan pemotongan *edge* sebesar 50% tidak membawa dampak pada hasil perangkingan "Kue Cubit". Perubahan hasil perangkingan ini terjadi karena dua faktor, yang pertama adalah hilangnya interaksi antar *node* hingga mengakibatkan terjadinya perubahan *shortest paths*. Faktor kedua adalah kemungkinan hilangnya seluruh *edge* dari suatu *node* terhadap *node* lain sebagai akibat rendahnya nilai *similarity* yang dimiliki, hal ini berimbas pada hilangnya kesempatan *node* tersebut untuk berperan sebagai *node* perantara didalam *shortest paths* serta hilangnya kemungkinan untuk berperan sebagai *node* sumber dan *node* tujuan.

Perubahan hasil perangkingan akibat pemotongan sejumlah *edge* hanya dapat diamati ketika k yang digunakan adalah k maksimal karena karakteristik hasil perangkingan k=100% yang selalu sama. Pengujian dengan k dibawah 100% seperti pada Tabel 4.5, Tabel 4.6, Tabel 4.7 tidak dapat menunjukkan pengaruh perubahan *edge* terhadap hasil perangkingan dengan jelas karena hasil pengujian lebih bergantung pada pemilihan *node* sumber daripada interaksi antar *node*. Pada pengujian dengan k=100% ketergantungan terhadap k dapat dianggap tidak berpengaruh karena semua *node* pernah menjadi *node* sumber dan yang berperan adalah interaksi antar *node*.

Berdasarkan hasil pengujian dengan menggunakan 4 model graf dapat dilihat bahwa jumlah *edge* mempengaruhi hasil *ranking betweenness centrality* dan dapat diamati dengan jelas pada nilai k maksimal. Perbedaan interaksi dalam suatu graf dengan *node* yang sama dapat menghasilkan perangkingan yang berbeda, perbedaan tersebut diakibatkan oleh 2 faktor yaitu perubahan *shortest paths* dan kemungkinan adanya sejumlah *node* yang tidak memiliki interaksi ke seluruh *node* lain karena terkena pemotongan *edge*.

4.6 Analisis Pengaruh Konten Terhadap *User*

Analisis pengaruh konten terhadap *user* dilakukan dengan membandingkan nilai *similarity* antar *node* menggunakan VSM. Dataset diambil dari 100 *user* dengan jumlah konten terbanyak dari tiap kata kunci. Jika *user* memiliki lebih dari satu konten maka konten tersebut akan digabung menjadi satu dokumen. Total *user* yang diperoleh dari 7 kata kunci adalah 534 *user*. Kata kunci yang dianalisis diantaranya adalah "Indomie", "Martabak", "Selai", "Brownies", "Kue Cubit", "@jkt66office" dan "Greentea".

Tabel 4.9 Dataset Konten

Kata Kunci	Jumlah *User*
Indomie	100
Martabak	100
Selai	100
Brownies	89
Kue Cubit	100
@jkt66office	36
Greentea	100

Tahap *preprocessing* dilakukan dengan memanfaatkan *library* lucene. Hasil *preprocessing*

kemudian digunakan sebagai masukan pada tahap penghitungan *similarity* menggunakan VSM. Keluaran dari penghitungan *similarity* dapat dibagi menjadi dua kelompok, yaitu kelompok tinggi [0.5-1] dan rendah [0-0,49].

Tabel 4.10 Nilai *Similarity* [0,5-1]

User 1	User 2	Nilai Similarity	Tweet User 1	Tweet User 2
meykeinemustopa	SofinaF	1	Suka"@Tweetnesian: suka Brownies ?"	Suka"@Tweetnesian: suka Brownies ?"
FitraNoviandara	ginaadestyanti	0.749412	Lg pengen cheeseCake dong :< "@Tweetnesian: Brownies / CheeseCake / Blackforest ?"	Blackforest "@Tweetnesian: Brownies / CheeseCake / Blackforest ?"
mariadpertiwi	Alifmarchs	0.687563	Tuh le"@infobandung: kue cubit green tea @JKT66official endesss banget. Bisa ditambah nutella atau toblerone lagi ?? http://t.co/bfp0P04B5m"	Gemess"@infobandung: kue cubit green tea @JKT66official endesss banget. Bisa ditambah nutella atau toblerone lagi ?? http://t.co/J6GO9YO9Tc"
SofinaF	Viviii_W	0.670703	Suka "@Tweetnesian: suka Brownies ?"	iyaaa"@Tweetnesian: suka Brownies ?"iyaaa jugaa"@Tweetnesian: suka Brownies ?"
achlulfadhlan	IvanFaadilla	0.605874	At Martabak Black Brownies Riung Bandung — https://t.co/mf8mbSQADH	With Mama at Martabak Black Brownies Riung Bandung — https://t.co/pOwd3auRty
noVindhaMarshel	KykaSetia	0.57357	At Outlet Brownies Amanda Dago Bandung — https://t.co/QjkHhnG5Cr	At Outlet Brownies Amanda Dago Bandung — https://t.co/HtqeGMugss

Berdasarkan Tabel 4.10 diketahui bahwa terdapat pola tertentu yang menyebabkan nilai *similarity* dari dua konten cukup tinggi. Pola pertama terjadi ketika terdapat dua *user* melakukan *retweet* dari suatu *tweet* yang sama. Pola kedua terjadi ketika kedua *user* melakukan *quote retwee* dari retweet yang sama, dimana selain melakukan *retweet* juga menambahkan kata atau kalimat tertentu. Pola terakhir yaitu ketika kedua *user* melakukan *posting* lokasi atau *mention* yang sama. Ketiga pola tersebut memiliki karakteristik khusus yaitu adanya kesamaan kata atau kalimat pada kedua konten, hal ini mengakibatkan penghitungan nilai VSM menunjukkan nilai yang mendekati atau sama dengan 1.

Tabel 4.11 Nilai *Similarity* [0-0,49]

User 1	User 2	Nilai Similarity	Tweet User 1	Tweet User 2
M_FakhriRamadan	sellayunii	0.494678699	Apa aku harus jadi INDOMIE biar jadi seleramu :(Blog jadi hayang ninyuh indomie goreng kadal weh	apa aku harus menjadi indomie agar menjadi seleramu :p ngakak aih
AWBasofi12	irasoisbrina	0.383615907	Martabak	Martabak ovaltine + martabak greentea-. Yipiiy-?????? (with Geldy at Martabak Inti Sari) [pic] — https://t.co/Nbe94dILsUOvaltine + kitkat greentea- Not Bad ?? @ martabak intisari http://t.co/q0JKsevYU6
dessymeyriana	ditaayuraa	0.281054978	With Atsa at Kue Cubit Asiong — https://t.co/t1QPn2Yz0v	At Kue Cubit Asiong — https://t.co/JD9LYSI3Wy
annisandryani	mayakaputri	0.149049327	Ngantri kue cubit green tea nya lama bgt deh ???? (with Silmina at Kue Cubit Asiong) — https://t.co/kmpZQG6K05 Kue cubit green tea ?? (at Kue Cubit Cisangkuy) [pic] — https://t.co/BleAHqZLZT	Kue cubit & ultra milk???? (at Taman Cikapayang) [pic] — https://t.co/ccvSiubJkq
mayakaputri	SabilaNfs	0.11602347	Kue cubit & ultra milk???? (at Taman Cikapayang) [pic] — https://t.co/ccvSiubJkq	Kue cubit greentea biar kekinian?? (with May at Kue Cubit Asiong) — https://t.co/hlBDSFdrBI
23Devinatalia	23Intanlestari	0	Akhirnya dapet juga kue ciwit green tea (at Kue cubit cisangkuy) — https://t.co/xI6qFpDuu3	Komplit rasanya kaya indomie rasa soto campur telor,kikil,keju,stoberi,jeruk pokoknya ga karuan~

Tabel 4.11 menunjukkan bahwa tidak semua interaksi antar konten memiliki nilai *similarity* yang tinggi. Nilai *similarity* antar dua konten juga dapat bernilai 0, hal ini terjadi karena adanya suatu pola tertentu dari dua dokumen yang dibandingkan. Jika kedua konten yang dibandingkan sama sekali tidak memiliki kata yang sama maka hasil penghitungan *similarity* dengan VSM akan menghasilkan nilai 0. Selain nilai 0 banyak juga hasil interaksi antar konten yang memiliki nilai sangat rendah seperti 0.116, dan 0.149. Nilai *similarity* yang sangat rendah namun bukan 0 menunjukkan bahwa kedua konten tersebut hanya mirip pada beberapa kata saja terutama pada kata kunci seperti “Indomie”, “Martabak”, dll.

Nilai *similarity* yang tinggi pada interaksi dua konten tidak serta merta membuat *user* pemilik konten tersebut menempati peringkat satu pada setiap perangkingan *betweenness centrality*. Hasil perangkingan tidak hanya dipengaruhi oleh nilai *similarity* antar *node* namun juga dipengaruhi oleh

centrality measurement yang digunakan, dalam hal ini *betweenness centrality*. *User* yang menempati peringkat pertama dalam perangkingan dengan metode *linear scaling* adalah *user* yang sering muncul dalam jalur *shortest paths* karena *user* tersebut memiliki nilai *similarity* yang tinggi terhadap konten dari *user* lain.

4.7 Hasil pengujian

Dari analisis sebelumnya berdasarkan sudut pandang hasil perangkingan maka dalam pengujian ini nilai k yang digunakan oleh setiap *kata kunci* berlainan dengan pertimbangan bahwa nilai k<<N dan hasil perangkingan mendekati perangkingan dengan nilai k maksimal. Pengujian dilakukan 10 kali untuk setiap *kata kunci* dengan nilai k ideal masing-masing, dari 10 hasil pengujian dipilih 2 untuk dirata-ratakan.

Tabel 4.12 Nilai k ideal

No	*Kata kunci*	k	k(%)
1	Indomie	10	7.90%
2	Martabak	20	10%
3	Selai	1	9%
4	Brownies	10	7.90%
5	Kue Cubit	20	10%
No	Kata Kunci	k	K(%)
6	jkt66office	3	8.30%
7	Greentea	20	10%

Pemilihan nilai k ini sudah sesuai dengan keterangan Geisberger yang mengatakan bahwa nilai k harus sangat kecil dari jumlah total *node* karena penghitungan *betweenness centrality* dengan metode *Linear Scaling* dilakukan melalui pendekatan. Batas maksimal k yang diambil adalah 10% dari total *node*.

Tabel 4.13 Hasil Rata-Rata 10 Besar pada *Kata kunci* "Selai"

Rank	*User*name	Betweenness
1	ApietHafizh	133.9632
2	Rarasatii	34.48015364
3	AstitaCindy	12.04474
4	AmandaPrdt	0
5	HadziqShop	0
6	Naminauttri	0
7	Novaliarika	0
8	NoviarioRK_	0
9	puspitha_n	0
10	Vionnas	0

Berdasarkan Tabel 4.6 dapat dilihat bahwa menurut pengujian *kata kunci* "Selai" dengan k=1 didapati "ApietHafizh" memimpin dengan skor *centrality* tertinggi disusul oleh "Rarasatii" berada pada peringkat 2 dan "AstitaCindy" pada urutan ke 3. Peringkat 4 hingga 10 memiliki tidak memiliki nilai betweenness *centrality*, hal ini terjadi karena *node* tersebut tidak pernah menjadi *node* perantara pada *shortest paths*. Hasil rata-rata 10 besar *kata kunci* selain "selai" terdapat di lampiran.

Tabel 4.14 *Username* dengan *Betweenness Centrality* Tertinggi

Kata kunci	*User*name	*Betweenness Centrality* (avg)
indomie	Rikaayulianti	21452952.07
martabak	audreyfaustina	1465226962
selai	ApietHafizh	133.9632
brownies	Firdharos	123042452.7
kue cubit	Yuliahaji	87889869
jkt66official	Prama_Prast	13447.61
greentea	putriyuniani	194299418.4

Berdasarkan Tabel 4.7 dapat dilihat bahwa setiap *kata kunci* memiliki *user* dengan nilai *betweenness*

centrality. Peringkat satu dari hasil perangkingan tiap *kata kunci* memiliki nilai *betweenness centrality* yang berlainan. Perbedaan nilai *centrality* pada peringkat satu ini disebabkan oleh perbedaan jumlah *node* yang ada pada masing-masing *kata kunci* dan frekuensi *node* tersebut terpilih sebagai *node* perantara pada *shortest paths*.

5. Kesimpulan & Saran

Berdasarkan hasil pengujian dan analisis yang sudah dilakukan maka dapat ditarik beberapa kesimpulan bahwa metode *linear scaling* dapat digunakan pada data Twitter untuk menentukan *ranking user* yang berpengaruh berdasarkan konten. Penghitungan nilai *betweenness centrality* pada kasus SNA berbasis konten dengan metode *linear scaling* dipengaruhi oleh nilai k, Semakin besar nilai k maka hasil perangkingan semakin tepat dan stabil, namun nilai k harus lebih kecil dari jumlah total *node* yang ada karena penghitungan *centrality* dengan metode *Linear Scaling* menggunakan pendekatan beberapa *node* saja. Jumlah *node* dalam suatu graf juga berpengaruh terhadap hasil penghitungan nilai *betweenness centrality.* Dari kesimpulan yang sudah didapatkan maka saran yang dapat diberikan dari penelitian ini dengan menggabungkan social network analysis berbasis konten dengan berbasis user dalam menghitung nilai betweenness centrality dan melakukan uji validitas hingga level user guna menentukan keabsahan hasil perangkingan.

Daftar Pustaka

[1] A. Culotta, "Detecting Influenza Outbreaks by Analyzing Twitter Messages," dalam *Department of Computer Science Southeastern Lousiana University*, Hammond, LA 70402, 2010.

[2] C. Streeter dan D. F. Gillespie, "Social Network Analysis," *Journal of Social Service Research*, no. 16, pp. 201-222, 1992.

[3] E. E. Santos, E. D. Sotelino, Y. Cao, E. Brown dan E. Santos, "Effective and Efficient Methodologies for Social Network Analysis," Computer Science and Applications, Balcksburg, Virginia, USA, 2007.

[4] Indonesia, Republik; Undang-Undang No.9 Tahun 1995 Tentang Usaha Kecil Menengah, Jakarta, 1995.

[5] J. Scoot, Social Network Analysis Theory and Application, Great Britain: Atheaum Press Ltd., Gateshead, Tyne & Wear, 2001.

[6] K. Borau, C. Ullrich, J. Feng dan S. R. , "Microblogging for Language Learning: Using Twitter to Train Communicative and Cultural Competence," Shanghai Jiao Tong University, Shanghai, 2009.

[7] L. C. Freeman, "*Centrality* in Social Networks Conceptual Clarification," *Social Networks*, vol. 1, pp. 215-239, 1979.

[8] O. Serrat, "Social Network Analysis," Cornell University ILR School, 2009.

[9] Otte, E and R. Rosseau, "Social Network Analysis: a Powerful Strategy, also for the Information Science," *Journal of Information Science*, vol. 28, pp. 443-455, 2002.

[10] P. D. Turney dan P. Pantel, "From Frequency to Meaning : Vector Space Models of Semantics," *Journal of Artificial Intelligence Research* , vol. 37, pp. 141-188, 2010.

[11] P. Velardi, N. Roberto, A. Cucchiarelli dan D. Fulvio, "A New Content-Based Model for Social Network Analysis," *IEEE International*

Conference on Semantic Computing, vol. icsc, pp. pp.18-25, 2008.

[12] R. Geisberger, "Better Approcimation of *Betweenness Centrality*," Universität Karlsruhe (TH), 2008.

[13] R. Geisberger, P. Sanders dan D. Schultes, "Better Approximation of *Betweenness Centrality*".

[14] S. Kumar, F. Morstatter dan H. Liu, Twitter Data Analytics, SpringerBriefs in Computer Science, 2013.

[15] U. Brandes, "A Faster Algorithm for *Betweenness Centrality*," *Journal of Mathematical Sociology* , vol. 25, no. 2, pp. 163-177, 2001.

[16] V. Krebs, "How to do Social Network Analysys," 2006. [Online]. Available: http://www.orgnet.com/sna.html.