Self Organizing Map-Neural Network untuk Pengelompokan Abstrak

Self Organizing Map - Neural Network for Abstract Clustering

Fajar Rohman Hariri*[1], Danar Putra Pamungkas[2]
[1,2] Universitas Nusantara PGRI Kediri
E-mail: *[1] **dosendeso@gmail.com**, [2] danar.aflach@gmail.com

Abstrak

Data berukuran besar yang sudah disimpan jarang digunakan secara optimal karena kemampuan manusia yang terbatas untuk mengelolanya. Salah satu data berskala besar adalah data teks. Data teks memiliki fitur yang besar sehingga untuk mengolahnya memerlukan waktu komputasi yang besar pula. Proses clustering menggunakan metode Self Organizing Map dengan menerapkan reduksi dimensi pada tahap preprosesing. Metode ini diterapkan untuk mengelompokkan data tugas akhir mahasiswa Teknik Informatika Universitas Trunojoyo Madura. Dalam metode yang diusulkan, analisis morfologi dilakukan pada teks abstrak tugas akhir mahasiswa untuk menghasilkan vektor input dengan unsur term dari tugas akhir tersebut. Dari percobaan yang dilakukan, diperoleh hasil bahwa optimum cluster menghasilkan nilai rata-rata SSE = 0.01117.

Kata Kunci *— Data Mining, Clustering, Self Organizing Map*

Abstract

Large data that is stored used rarely optimally because of the limited human ability to manage it. One of large-scale data is text data. Text data has enormous features so as to process it requires greater computational time. Clustering process using Self Organizing Map by applying dimensionality reduction on preprocessing. This method is applied to cluster the Informatics Engineering students' final assignment data of Trunojoyo University. In the proposed method, morphological analysis is applied on the abstract of final assignment to generate input vectors using elements of the final assignment. From the experiments conducted, the result that the best cluster to abstract data, average value of SSE = 0.01117.

Keywords *— Data Mining, Clustering, Self Organizing Map*

1. PENDAHULUAN

Perkembangan teknologi telah mengakibatkan meningkatnya data dalam jumlah besar. Data berukuran besar yang sudah disimpan jarang digunakan secara optimal karena manusia seringkali tidak memiliki waktu dan kemampuan yang cukup untuk mengelolanya. Data bervolume besar seperti data teks, jauh melampaui kapasitas pengolahan manusia yang sangat terbatas [1].

Kasus yang disoroti adalah data tugas akhir mahasiswa UTM khususnya jurusan teknik informatika. Setiap tahun fakultas dan perpustakaan menerima dan menyimpan data penelitian tugas akhir mahasiswa. Namun data tersebut dibiarkan begitu saja tanpa adanya pengolahan atau tindak lanjut. Padahal data tersebut dapat dijadikan pembelajaran bagi pihak lainnya khususnya mahasiswa yang melakukan penelitian terkait.

Data mining sangat sesuai untuk diterapkan pada data berukuran besar. Penerapan data mining pada data tugas akhir mahasiswa UTM diharapkan bisa menambang ilmu pengetahuan dan informasi yang penting dan berguna untuk pengambilan keputusan di masa depan. Metode data mining yang akan diterapkan dalam penelitian ini adalah clustering atau pengelompokan data [2].

Pengelompokkan artikel atau teks berbahasa Indonesia merupakan salah satu solusi yang dapat digunakan untuk mempermudah mencerna informasi penting yang ada di dalamnya. Clustering dapat digunakan untuk membantu menganalisis berita dengan mengelompokkan secara otomatis berita yang memiliki kesamaan. Pada text clustering terdapat suatu permasalahan yaitu adanya fitur-fitur yang berdimensi tinggi sehingga menyebabkan waktu komputasi menjadi besar. Oleh karena itu dalam tugas akhir ini digunakan metode Self-Organizing Maps Neural Network (SOM-NN) dengan menerapkan pembatasan wilayah pencarian dan reduksi dimensi sehingga diharapkan dapat mengurangi waktu komputasi dalam proses clustering. Salah satu keunggulan dari algoritma Self-Organizing Map adalah mampu untuk memetakan data berdimensi tinggi ke dalam bentuk peta berdimensi rendah [3].

Penelitian yang berkaitan dengan *text clustering* skala besar telah dilakukan oleh Tetsuya Toyota dan Hajime Nobuhara pada tahun 2011. Dalam penelitiannya yang bertajuk "*Visualization of the Internet News Based on Efficient Self-Organizing Map Using Restricted Region Search and Dimensionality Reduction*" mengusulkan sebuah sistem untuk mengatasi masalah waktu komputasi yang besar ketika SOM berurusan dengan data berskala besar. Melalui percobaan dan evaluasi diketahui bahwa metode yang diusulkan untuk mempercepat SOM dapat mengurangi waktu komputasi turun menjadi sekitar 25% dari metode konvensional SOM pada umumnya.

2. METODE PENELITIAN

2.1. Metode Penelitian

Penelitian ini merupakan penelitian eksperimental, karena untuk mendapatkan performa terbaik algortima SOM-NN dalam clustering data, dilakukan beberapa kali percobaan dengan parameter yang berbeda-beda. Data didapatkan dari abstrak tugas akhir mahasiswa jurusan teknik informatika Universitas Trunojoyo Madura dari empat bidang minat. Dokumen abstrak tersebut dalam format Ms. Word yang diperoleh dari Kantor Prodi Teknik Informatika Universitas Trunojoyo Madura.

2.2. Metode Analisis Data

Terdapat dua jenis data yang akan digunakan dalam sistem ini, yaitu data testing dan training. Data training digunakan untuk mencari bobot akhir yang paling konstan, sehingga bobot tersebut nantinya akan diujikan pada data testing. Data tersebut adalah kumpulan abstrak tugas akhir mahasiswa jurusan teknik informatika dari empat bidang minat. Dokumen abstrak tersebut dalam format Ms. Word yang diperoleh dari Kantor Prodi Teknik Informatika Universitas Trunojoyo Madura.

Jumlah data training adalah 80 Dokumen, sedangkan untuk data testing adalah 60 dokumen. Pembagian jenis data akan dijelaskan pada tabel 1 dan tabel 2 berikut:

Tabel 1. Data Training

No.	Bidang Minat	Abstrak
1	CAI	20
2	SI/RPL	20
3	SISTER	20
4	Multimedia	20

Tabel 2. Data Testing

No.	Bidang Minat	Abstrak
1	CAI	22
2	SI/RPL	20
3	SISTER	7
4	Multimedia	11

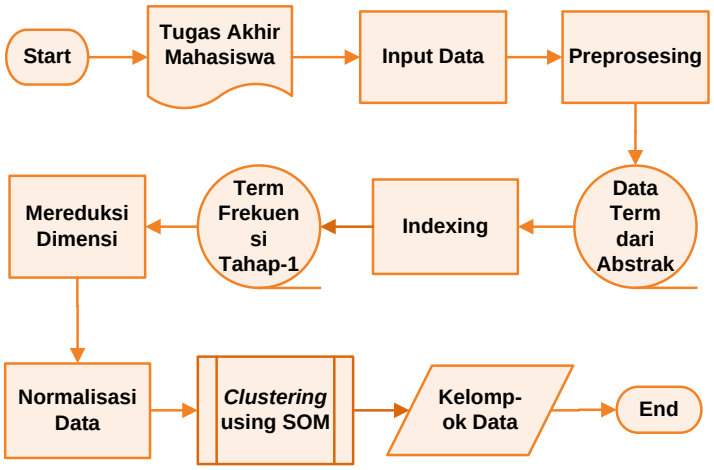
Terdapat 3 skenario ujicoba pada tiap data. Perbedaan atar skenario adalah jumlah term yang dipakai atau jumlah sisa term yang ada setelah proses reduksi. Berikut adalah tabel 3 yang menjelaskan skenario ujicoba:

Tabel 3. Jumlah term pada skenario ujicoba

No	Data	Skenario 1	Skenario 2	Skenario 3
1	Abstrak	150	44	27
2	Laporan	178	346	220

2.3. Alur Penelitian

Secara umum, dilakukan beberapa tahapan seperti pada gambar 1. :



Gambar 1. Alur Penelitian

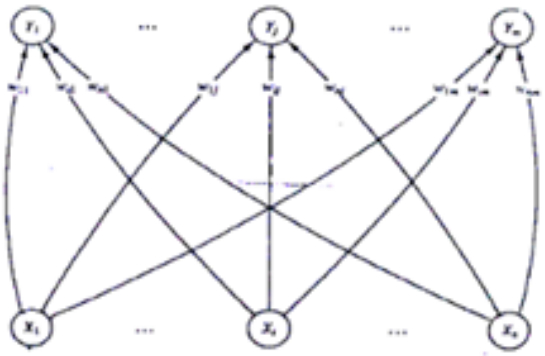
Alur penelitian dari penelitian ini adalah:

1) Input Data
   Mencari data-data yang terkait dengan sistem, antara lain mengumpulkan data tugas akhir yang berupa abstrak penelitian mahasiswa jurusan teknik informatika Universitas Trunojoyo Madura. Kemudian dimasukkan ke dalam sistem.
2) Preprocessing
   a. Stemming Algoritma Nazief & Andriani.
      Proses pemecahan kata dan mengambil kata dasar dari dokumen abstrak menggunakan Algoritma Nazief & Andriani.
3) Indexing
   Dari data kata dasar masing-masing dokumen akan dilist dan dilihat frekuanesi masing-masing kata untuk selanjutnya dibentuk sebuah matrix *term frequent.*
4) Reduksi Dimensi
   Dokumen-dokumen abstrak yang ada akan menghasilkan banyak kata dasar, dan ada beberapa kata yang hanya akan muncul di satu atau sedikit dokumen, sehingga untuk selanjutnya akan dilakukan proses reduksi dimensi dengan membuang kata-kata itu.

5) Normalisasi Data
   Setelah data di reduksi, kemudian data dinormalisasikan supaya range data tidak terlalu jauh.
6) Clustering menggunakan SOM-NN
   Setelah diterapkan metode SOM-NN terhadap data Term Frequent (TF), maka akan diperoleh bobot akhir yang paling konstan. Bobot tersebut akan digunakan sebagai bobot referensi untuk pengecekan data testing. Dalam penentuan testing dapat dilihat pada nilai Euclidian distance dari bobot akhir terhadap data baru. Yang paling minimumlah yang menjadi pemenangnya, dengan kata lain cluster tersebutlah yang menang.

2.4. Self-Organizing Maps

Metode Self-Organizing Maps (SOM-NN) bertujuan untuk mengklasifikasikan suatu vektor-vektor input berdasarkan bagaimana mereka mengelompok sesuai dengan karakteristik inputnya. Learning SOM-NN bekerja dengan cara menggabungkan proses competitive layers dengan topologi vektor-vektor input yang dimasukkan dalam proses iterasi. Jaringan SOM-NN terdiri dari dua lapisan (layer), yaitu lapisan input dan lapisan output. Setiap neuron dalam lapisan input terhubung dengan setiap neuron pada lapisan output. Setiap neuron dalam lapisan output merepresentasikan kelas dari input yang diberikan. Selama proses penyusunan diri, cluster yang memiliki vektor bobot paling cocok dengan pola input (memiliki jarak paling dekat) akan terpilih sebagai pemenang. Neuron yang menjadi pemenang beserta neuron-neuron tetangganya akan memperbaiki bobot-bobotnya. Apabila ingin membagi data-data menjadi K cluster, maka lapisan kompetitif akan terdiri atas K buah neuron. Arsitektur dari self-organizing map dapat dilihat pada gambar 2.



Gambar 2. Arsitektur Self-Organizing Maps

Algoritma pengelompokan pola jaringan SOM-NN adalah sebagai berikut:

a) Inisialisasi awal Bobot wij (random)
b) Nilai parameter learning rate (α) dan radius tetangga (R).
c) Selama kondisi penghentian bernilai salah, lakukan langkah 2 – 8
d) Untuk setiap input vektor x, (xi, i = 1, 2, ..., n) lakukan langkah 3 – 5
e) Hitung jarak Euclidian untuk semua j (j = 1, 2, ..., m)

$$D_{(j)} = \sum_{i=1}^{n} \left(w_{ij} - x_i\right)^2 \qquad (1)$$

f) Tentukan indeks j sedemikian sehingga D(j) minimum
g) Melakukan perbaikan wij dengan nilai tertentu, yaitu:

$$w_{ij\ (baru)} = w_{ij\ (lama)} + a\left[x_i - w_{ij\ (lama)}\right] \qquad (2)$$

h) Modifikasi parameter learning rate
i) Uji kondisi penghentian

Kondisi penghentian iterasi adalah selisih antara wij saat itu dengan wij pada iterasi sebelumnya. Apabila semua wij hanya berubah sedikit saja, berarti iterasi sudah mencapai konvergensi sehingga dapat dihentikan [4].

2.5. Reduksi Dimensi Berdasarkan Karakteristik Data Artikel

Proses reduksi dimensi dilakukan untuk mengecilkan dimensi data sehingga waktu komputasi dibutuhkan lebih sedikit. Namun proses reduksi dimensi harus memperhatikan karakteristik data, karena dimensi yang hilang bisa jadi juga menghilangkan karakteristik data[5]. Oleh karena itu dalam penelitian ini dipilih reduksi dimensi dengan menghilangkan *sparse part*. Term yang memiliki frekuensi yang kebanyakan bernilai nol akan dihilangkan, karena term tersebut merepresentasikan term yang sangat jarang dijumpai dalam data atau dengan kata lain term boleh diabaikan.

Besarnya pengurangan dimensi term sesuai dengan parameter yang ditentukan. Karena ditakutkan proses ini akan menghilangkan karakteristik data, maka dibuat parameter nilai yang berfungsi untuk membatasi reduksi dimensi. Sehingga meskipun dalam suatu term kebanyakan nilainya nol, term tersebut tidak akan dihilangkan jika ada salah satu dari nilai frekuensinya melebihi parameter yang telah ditentukan.[6]

Data TF yang sudah masuk dalam database memiliki nilai yang berbeda-beda. Term yang memilik TF yang kebanyakan bernilai nol dapat diabaikan, dengan catatan tidak ada nilai TF yang melebihi parameter yang sudah ditentukan. Dengan kata lain proses reduksi dimensi adalah dengan cara menghilangkan fitur atau term yang frekuensinya semua/kebanyakan bernilai 0. Jika ada beberapa yang tidak bernilai 0 maka TF tersebut nilainya harus melebihi parameter tertentu. Hal ini dimaksudkan agar data tidak kehilangan karakteristiknya. Sehingga proses dapat berjalan dengan baik. [6]

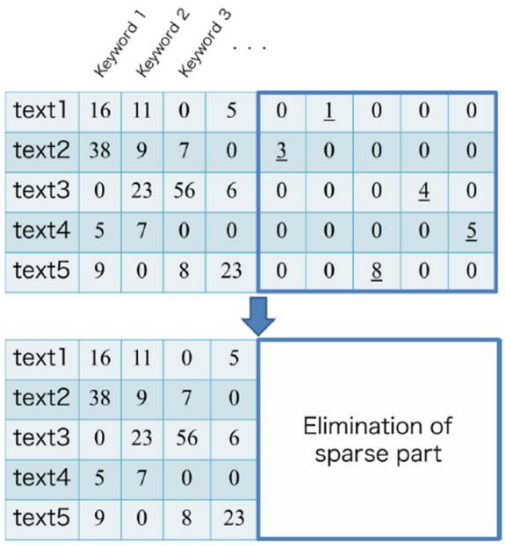


Fig. 3. Example of dimensionality reduction.

Gambar 3. Ilustrasi Reduksi Dimensi

Dalam gambar 3, ditunjukkan misal parameter adalah 40%. Maka reduksi dimensi dilakukan dengan menghilangkan kolom yang nilainya bukan 0 kurang dari 40% data yang ada. Kolom ke 5-9 akan dihilangkan karena kolom tersebut tidak memenuhi syarat.

2.6. Algoritma Nazief & Adriani

Algoritma stemming untuk bahasa yang satu berbeda dengan algoritma stemming untuk bahasa lainnya. Sebagai contoh bahasa Inggris memiliki morfologi yang berbeda dengan bahasa Indonesia sehingga algoritma stemming untuk kedua bahasa tersebut juga berbeda. Proses stemming pada teks berbahasa Indonesia lebih rumit/kompleks karena terdapat variasi imbuhan yang harus dibuang untuk mendapatkan root word (kata dasar) dari sebuah kata. Pada umumnya kata dasar pada bahasa Indonesia terdiri dari kombinasi: [7]

Prefiks 1 + Prefiks 2 + Kata dasar + Sufiks 3 + Sufiks 2 + Sufiks 1

Algoritma Nazief & Adriani yang dibuat oleh Bobby Nazief dan Mirna Adriani ini memiliki tahap -tahap sebagai berikut: (Nazief, 1996)

1. Pertama cari kata yang akan diistem dalam kamus kata dasar. Jika ditemukan maka diasumsikan kata adalah root word, maka algoritma berhenti.
2. Inflection Suffixes (“-lah”, “-kah”, “-ku”, “-mu”, atau “-nya”) dibuang. Jika berupa particles (“-lah”, “ -kah”, “-tah” atau “-pun”) maka langkah ini diulangi lagi untuk menghapus Possesive Pronouns (“-ku”, “-mu”, atau “-nya”) jika ada.
3. Hapus Derivation Suffixes (“-i”, “ -an” atau “-kan”). Jika kata ditemukan di kamus, maka algoritma berhenti. Jika tidak maka ke langkah 3a
   a. Jika “-an” telah dihapus dan huruf terakhir dari kata tersebut adalah “-k”, maka “-k” juga ikut dihapus. Jika kata tersebut ditemukan dalam kamus maka algoritma berhenti. Jika tidak ditemukan maka lakukan langkah 3b.
   b. Akhiran yang dihapus (“-i”, “-an” atau “-kan”) dikembalikan, lanjut ke langkah 4.
4. Hilangkan derivation prefixes DP {“di-”,“ke-”,“se-”,“me-”,“be-”,“pe”, “te-”} dengan iterasi maksimum adalah 3 kali:
   a. Langkah 4 berhenti jika:
   - Terjadi kombinasi awalan dan akhiran yang terlarang seperti pada Tabel 4.
   - Awalan yang dideteksi saat ini sama dengan awalan yang dihilangkan sebelumnya.
   - Tiga awalan telah dihilangkan.

Tabel 4. Kombinasi Awalan dan Akhiran yang tidak diijinkan

Awalan	**Akhiran yang tidak diijinkan**
be-	-i
di-	-an
ke-	-i , -kan
me-	-an
se-	-i , -kan
te-	-an

   b. Identifikasikan tipe awalan dan hilangkan. Awalan ada tipe :
   - Standar: “di-”, “ke-”, “se-” yang dapat langsung dihilangkan dari kata.
   - Kompleks: “me-”, “be-”, “pe”, “te-” adalah tipe-tipe awalan yang dapat bermorfologi sesuai kata dasar yang mengikutinya. Oleh karena itu, gunakan aturan pada Tabel 5 untuk mendapatkan pemenggalan yang tepat.

Tabel 5. Aturan awalan me-,be-,pe-,te-

Aturan	Format Kata	Pemenggalan
1	berV..	Ber-V | ber-rV
2	berCAP…	Ber-CAP dimana C !=’r’ & P!=’er’
3	berCAerV..	berCaerV dimana C !=’r’
4	Belajar	Bel-ajar
5	beCerC	Be-CerC dimana C !=’r’|’l’
6	terV	Ter-V | te-rV
7	terCerV	Ter-CerV dimana C !=’r’
8	terCP..	Ter-CP dimana C !=’r’ & P!=’er’
9	teCerC..	Te-CerC dimana C !=’r’
10	Me{l|r|w|y}V..	Me-{l|r|w|y}V..
11	Mem{b|f|v}..	Mem-{b|f|v}..
12	Mempe{l|r}	Mem-pe{l|r}
13	Mem{rV|V}	Me-m{rV|V} | Me-p{rV|V}
14	Men{c|d|j|z}..	Men-{c|d|j|z}..

Tabel 5. (Lanjutan)

Aturan	Format Kata	Pemenggalan
15	MenV	Me-nV | me-tV
16	Meng{g|h|q}	Meng-{g|h|q}
17	mengV	Meng-V | Meng-kV
18	menyV	Meny-sV
19	mempV	Mem-pV dimana V!='e'
20	Pe{w|y}V	Pe-{w|y}V
21	perV	Per-V | pe-rV
22	perCAP	Per-CAP dimana C !='r' & P!='er'
23	perCaerV	Per-CaerV dimana C !='r'
24	pem{b|f|v}..	Pem-{b|f|v}..
25	pem{rV|V}	Pe-m{rV|V} | Pe-p{rV|V}
26	pen{c|d|j|z}..	Pen-{c|d|j|z}..
27	penV	Pe-nV | Pe-tV
28	Peng{g|h|q}	Peng-{g|h|q}
29	pengV	Peng-V | peng-kV
30	penyV	Peny-sV
31	pelV	Pe-lV kecuali ajar yang menghasilkan ajar
32	peCerV	peC-erV dimana C!={r|w|y|l|m|n}
33	peCP	Pe-CP dimana C!={r|w|y|l|m|n} dan P!='er'

b. Cari kata yang telah dihilangkan awalannya ini di dalam kamus. Apabila tidak ditemukan, maka langkah 4 diulangi kembali. Apabila ditemukan, maka keseluruhan proses dihentikan.

5. Apabila setelah langkah 4 kata dasar masih belum ditemukan, maka proses recoding dilakukan dengan mengacu pada aturan pada Tabel 5. Recoding dilakukan dengan menambahkan karakter recoding di awal kata yang dipenggal. Pada Tabel 5, karakter recoding adalah huruf kecil setelah tanda hubung ('-') dan terkadang berada sebelum tanda kurung. Sebagai contoh, kata "menangkap" (aturan 15), setelah dipenggal menjadi "nangkap". Karena tidak valid, maka recoding dilakukan dan menghasilkan kata "tangkap".
6. Jika semua langkah telah selesai tetapi tidak juga berhasil maka kata awal diasumsikan sebagai root word dan proses selesai.

Tipe awalan ditentukan melalui langkah-langkah berikut:

1. Jika awalannya adalah: "di-", "ke-", atau "se-" maka tipe awalannya secara berturut-turut adalah "di-", "ke-", atau "se-".
2. Jika awalannya adalah "te-", "me-", "be-", atau "pe-" maka dibutuhkan sebuah proses tambahan untuk menentukan tipe awalannya.
3. Jika dua karakter pertama bukan "di-", "ke-", "se-", "te-", "be-", "me-", atau "pe-" maka berhenti.
4. Jika tipe awalan adalah "none" maka berhenti. Jika tipe awalan adalah bukan "none" maka awalan dapat dilihat pada Tabel 6. Hapus awalan jika ditemukan.

Untuk aturan penghapusan awalan ada pada tabel 6 dan tabel 7.

Tabel 6. Cara menentukan tipe awalan "te-"

Following Character				Tipe Awalan
Set 1	Set 2	Set 3	Set 4	
"-r-"	"-r-"	-	-	None
"-r-"	-	-	-	Ter-luluh
"-r-"		"-er-"	Vowel	Ter
"-r-"		"-er-"	Not vowel	Ter-
"-r-"		Not "-er-"	-	Ter
Not (vowel or "r")	"-er-"	Vowel	-	none
Not (vowel or "r")	"-er-"	Not vowel	-	None

Tabel 7. Jenis awalan berdasarkan tipe awalannya

Tipe awalan	Awalan yang harus dihapus
Di-	Di-
Ke-	Ke-
Se-	Se-
Te-	Te-
Ter-	Ter-
Ter-luluh	Ter-

Untuk mengatasi keterbatasan pada algoritma di atas, maka ditambahkan aturan-aturan di bawah ini:

1. Aturan untuk reduplikasi.
   a. Jika kedua kata yang dihubungkan oleh kata penghubung adalah kata yang sama maka root word adalah bentuk tunggalnya, contoh :
   "buku-buku" root word-nya adalah "buku".
   b. Kata lain, misalnya "bolak-balik", "berbalas-balasan, dan "seolah-olah". Untuk mendapatkan root word-nya, kedua kata diartikan secara terpisah. Jika keduanya memiliki root word yang sama maka diubah menjadi bentuk tunggal, contoh: kata "berbalas-balasan", "berbalas" dan "balasan" memiliki root word yang sama yaitu "balas", maka root word "berbalas-balasan" adalah "balas". Sebaliknya, pada kata "bolak-balik", "bolak" dan "balik" memiliki root word yang berbeda, maka root word-nya adalah "bolak-balik".
2. Tambahan bentuk awalan dan akhiran serta aturannya.
   a. Untuk tipe awalan "mem-", kata yang diawali dengan awalan "memp-" memiliki tipe awalan "mem-".
   b. Tipe awalan "meng-", kata yang diawali dengan awalan "mengk-" memiliki tipe awalan "meng-".

3. HASIL DAN PEMBAHASAN

3.1. Implementasi

Proses yang dilakukan setelah tahap reduksi dimensi adalah proses training atau learning, yaitu dengan menerapkan metode SOM-NN pada data training. Setelah maximum iterasi tercapai, didapatkan bobot akhir yang paling konstan. Bobot tersebut yang nantinya digunakan untuk proses uji coba terhadap data testing.

Pada proses testing tidak lagi menerapkan metode SOM-NN, melainkan hanya dengan mengukur jarak antara data testing dengan bobot akhir. Proses ini disebut dengan menghitung jarak Euclidean atau Euclidean distance. Setelah diketahui pada bobot ke-n adalah jarak paling kecil yang didapat terhadap data testing ke-m, maka hasil cluster menunjukkan data ke-m masuk dalam cluster n.

Uji coba dilakukan dengan menggunakan tiga skenario seperti pada tabel 8 berikut:

Tabel 8. Skenario Ujicoba

Skenario	Reduksi Dimensi
1	• Batas jumlah 0 pada satu kolom < 80%. • Batas value TF minimum > 5 term. • Jumlah term yang dihasilkan = 150 term.
2	• Batas jumlah 0 pada satu kolom < 80%. • Batas value TF minimum > 50 term. • Jumlah term yang dihasilkan = 44 term.
3	• Batas jumlah 0 pada satu kolom < 70%. • Batas value TF minimum > 50 term. • Jumlah term yang dihasilkan = 27 term.

3.2. Hasil Ujicoba

Untuk mengevaluasi hasil cluster dilakukan dengan menghitung Sum Squared Error (SSE). SSE merupakan jumlah kuadrat perbedaan antara observasi dengan rata-rata perklaster. Hal ini dapat digunakan sebagai ukuran variasi dalam sebuah klaster. Jika semua kasus dalam sebuah klaster adalah identik maka nilai dari SSEnya sama dengan 0 [8]. Berikut persamaan SSE yang akan digunakan:

$$SSE = \sum_{i=1}^{n} (x_i - \bar{x})^2 \quad (3)$$

Keberhasilan Hasil Cluster diukur menggunakan SSE. SSE dihitung pada tiap cluster dalam tiap skenario. Semakin kecil nilai SSE maka semakin baik hasil clusternya. Optimum cluster terdapat pada skenario yang memiliki nilai rata-rata SSE.

Berikut adalah tabel 9 yang menunjukkan nilai SSE tiap cluster dalam tiap skenario:

Tabel 9. Rata-rata SSE

No	Skenario	Cluster	Nilai SSE	Rata-rata SSE
1	1	1	0.0183	0.011175
2		2	0.0042	
3		3	0.0127	
4		4	0.0095	
5	2	1	0.0161	0.077025
6		2	0.0121	
7		3	0.0059	
8		4	0.2740	
9	3	1	0.0156	0.0148
10		2	0.0244	
11		3	0.0115	
12		4	0.0077	

Dari tabel di atas dapat diketahui bahwa nilai rata-rata SSE terkecil terdapat pada skenario 1 dengan jumlah term 150, nilai rata-rata SSE 0.011175 dan waktu komputasi sebesar 932 detik.

4. KESIMPULAN

Dari hasil percobaan dapat diketahui bahwa penelitian tentang Implementasi Efficient Self-Organizing Map pada Text Clustering menggunakan reduksi dimensi dapat diterapkan pada data tugas akhir mahasiswa teknik informatika. Percobaan dilakukan dengan mengcluster tugas akhir tersebut menjadi empat kelompok, karena pada dasarnya data tugas akhir tersebut terdiri dari empat bidang minat.

Ujicoba dilakukan dengan cara menguji data melalui beberapa skenario percobaan. Skenario percobaan dibagi menjadi 3 skenario. Optimum cluster terdapat pada skenario 1 yaitu dengan nilai SSE = 0.011175 dan waktu komputasi sebesar 923 detik.

Dapat disimpulkan bahwa optimum cluster untuk data abstrak adalah dengan jumlah fitur 150 term. Proses reduksi dimensi sangat mempengaruhi waktu komputasi, sehingga bisa diterapkan SOM-NN pada data tugas akhir mahasiswa universitas trunojoyo.

5. SARAN

Pada penelitian selanjutnya diharapkan para peneliti bisa menggunakan metode parseWord yang lebih baik, sehingga term yang tersaring dari dokumen Ms.Word lebih banyak dan bagus. Selain itu memungkinkan apabila selanjutnya dokumen yang diambil sebagai data bukan dokumen Ms. Word melainkan dengan format PDF. Jika sistem bisa membaca langsung dari dokumen PDF maka sistem akan lebih baik.

DAFTAR PUSTAKA

[1] Edward., 2007, Clustering Menggunakan Self-Organizing Maps (Studi Kasus: Data PPMB IPB), *Skripsi*, FMIPA, Institut Pertanian Bogor, Bogor.

[2] Toyota, T., Nobuhara, H., 2012, Visualization of the Internet News Based on Efficient Self-Organizing Map Using Restricted Region Search and Dimensionality Reduction, *JACIII*, No. 2, Vol. 12, Hal 219-226.

[3] Langgeni, D.P., Baizal, Z. K. A., Firdaus, Y., 2010, Clustering Artikel Berita Berbahasa Indonesia Menggunakan Unsupervised Feature Selection, Seminar Nasional Informatika 2010 (semnasIF), Yogyakarta, 22 Mei 2010.

[4] Kusumadewi, S., 2004, *Membangun Jaringan Syaraf Tiruan (menggunakan MATLAB & Excel Link)*, Penerbit Graha Ilmu, Yogyakarta.

[5] Nazief, B., Adriani, M., 1996. Confix-Stripping: Approach to Stemming Algorithm for Bahasa Indonesia, Laporan Penelitian, Fakultas Ilmu Komputer, Universitas Indonesia.