Jurnal Ilmiah KURSOR
Menuju Solusi Teknologi Informasi

**Vol. 5, No. 4, Juli 2010** **ISSN 0216 - 0544**

# KLASTERISASI TEKS MENGGUNAKAN METODE *MAX-MAX ROUGHNESS* (*MMR*) DENGAN PENGAYAAN SIMILARITAS KATA

**Mohammad Rizal Arief, Daniel O Siahaan, Isye Arieshanti**
Program Pasca Sarjana, Jurusan Teknik Informatika, ITS
Jl. Raya ITS, Kampus ITS, Sukolilo, Surabaya, 60111
E-Mail: rizal_st@cs.its.ac.id

***Abstrak***

Klasterisasi teks mempunyai salah satu permasalahan utama dalam mengklasifikasikan jenis teks yang mempunyai sifat uncertain atau sulit dikategorikan pada data berdimensi yang tinggi dan menyebar. Pada penelitian ini diperkenalkan metode baru dalam penyusunan klasterisasi teks berbasis *Roughset* untuk persamaan kata. Metode yang diusulkan dalam penelitian ini bernama *Max-max Roughness* (*MMR*). *Roughset* dipilih karena terbukti mampu mengatasi permasalahan *data uncertain*. Metode klasterisasi teks umumnya dilakukan dengan mencari persamaan dokumen berdasarkan bobot semua kata dalam masing-masing dokumen (perbandingan obyek), sedangkan metode ini menggunakan beberapa kata kunci yang mempunyai perwakilan paling besar (*max roughness*) dalam dokumen untuk proses klasterisasi (perbandingan atribut). Metode *MMR* ini menyertakan proses *enriching* (pengayaan) pada dokumen. Proses uji coba dan evaluasi pada penelitian ini dilakukan dengan menggunakan data 20 *News groups*, yang telah dianalisa menggunakan teori *K-Means* dengan pembobotan (*FW-Kmeans*). Pengukuran hasil pengujian dilakukan menggunakan teori *entropy* klaster. Hasil yang diperoleh menunjukkan bahwa metode ini dapat meningkatkan kualitas hasil klasterisasi rata-rata sebesar 30,28% dibandingkan hasil metode *K-Means* dengan pembobotan.

Kata kunci*: K-Means, Uncertain, Roughness, MMR, entropy.*

***Abstract***

*Clustering of text is one of the main problems in classifying kinds of texts having uncertain characters and difficult to be categorized for data dimension which are high and sporadic. This research discusses a new method to arrange text clustering based on Roughset for words similarity. This method is named Max-max Roughness (MMR), for this method is able to handle uncertain data problems. Clustering method is done generally by observing the document similarity based on the value of all words in every documents (object comparison), besides this method also uses several key words having the highest representation (max Roughness) in the document for clustering process (attribute comparison). MMR method also includes enriching process in document. Process of examination and evaluation in this research is done with 20 data news group, which have been analyzed with K-Means theory using weighting (FW-Kmeans). The measurement of experiment result was done with entropy cluster theory. The result show that this method can increase the quality of clustering result in average 30,28% compared with K-Means method with weighting.*

*Key words: K-Means, Uncertain, Roughness, MMR, entrophy.*

## PENDAHULUAN

Teks merupakan sarana interaksi dalam semua media komunikasi tulisan. Oleh karena peningkatan ukuran dan jenisnya sangatlah cepat, maka analisa data teks menjadi sesuatu yang sangat penting. Penggalian teks menjelma menjadi teknologi yang bersifat darurat. Klasterisasi teks merupakan salah satu fungsi fundamental dalam penggalian teks. Klasterisasi teks sangat berpengaruh dalam penggalian teks karena menunjukkan topik yang terdapat dalam dokumen dan mengidentifikasikan kata kunci dari setiap topik. Beberapa fungsi dari klasterisasi teks adalah untuk pengelompokan *web searching* dan penyusunan kategori dokumen digital.

Klasterisasi teks berhubungan dengan jumlah data teks yang besar, dimensi yang tinggi, dan struktur yang terus berubah. Pada penggalian teks, data yang digunakan umumnya berbentuk tidak terstruktur atau minimal semi terstruktur. Jenis data yang tidak terstruktur atau semi terstruktur tersebut menyebabkan adanya permasalahan pada klasterisasi teks karena algoritma yang digunakan dalam klasterisasi hanya mendukung terhadap teks yang bersifat terstruktur. Transformasi teks tidak terstruktur menjadi bentuk terstruktur dapat dilakukan dengan menggunakan *Vector Space Model* (*VSM*). Dalam *VSM*, data direpresentasikan ke dalam vektor, yaitu jumlah baris menunjukkan banyaknya dokumen dalam data set, sedangkan jumlah kolom mewakili banyaknya kata yang terdapat dalam *dataset* tersebut.

Beberapa penelitian dalam klasterisasi data teks telah dilakukan [1][2][3]. Penelitian dengan menggunakan *K-Means* berbasis *Roughset* menunjukkan pembentukan klaster berdasarkan similaritas antar dokumen. Klaster dibentuk dengan menghitung semua bobot kata yang mempunyai persamaan pada masing-masing dokumen (perbandingan obyek). Keterbatasan dari perbandingan antar obyek terletak dalam penentuan dokumen lain yang digunakan sebagai pembanding dalam menentukan similaritas antar dokumen. Bila atribut dokumen (kata) yang dibandingkan antar dokumen sangat sedikit, dokumen dalam kelas yang sama dianggap tidak mempunyai persamaan. Untuk mengatasi hal tersebut dilakukan pembagian klaster terlebih dahulu (*subspace clustering*) dengan mencari kata kunci yang dapat mewakili dokumen ke dalam klaster [4]. Metode yang digunakan adalah dasar *K-Means* pembobotan (*FW-Kmeans*) dengan perbandingan obyek pula. Keterbatasan yang dimiliki adalah bahwa ia bersifat dan tidak memperhitungkan kata-kata dalam dokumen lain yang berhubungan dengan dokumen itu, tetapi tidak terdapat dalam dokumen tersebut, dan mengeliminasi 15% dari kata kunci karena dianggap tidak merepresentasikan dokumen ke dalam klaster.

Penelitian lain dalam klasterisasi dilakukan terhadap data kategorikal dan bukan teks [5]. Berbeda dengan klasterisasi sebelumnya yang menggunakan *K-Means* dan bersifat antar obyek, klasterisasi ini menggunakan *roughset* murni sebagai dasar perhitungannya dan bersifat antar atribut. Obyek penelitian dalam klasterisasi ini adalah data *supplier*, sehingga metode pembobotan dan pengayaan tidak dilakukan, seperti halnya pada obyek teks.

Pada penelitian ini diusulkan sistem klasterisasi baru pada penggalian teks berdasarkan perhitungan kata yang mempunyai *Roughness* paling tinggi pada seluruh dokumen (perbandingan atribut) dengan pengayaan similaritas. Sistem klasterisasi ini diharapkan dapat menyelesaikan permasalahan *uncertainty* tanpa melakukan eliminasi pada kata kunci pada penelitian sebelumnya. Berbeda dengan penelitian sebelumnya yang menggunakan perbandingan obyek atau dokumen dalam penyusunan klaster, metode baru ini menggunakan perbandingan antar atribut (kata) dalam melakukan proses klasterisasi teks. Sistem ini terdiri dari dua proses utama, yaitu representasi model (*preprocessing*, pembobotan atau *weigthing*, aproksimasi atas, serta diskritisasi) dan klasterisasi.

Pemrosesan awal dilakukan untuk mendapatkan *VSM* data dokumen. Pemrosesan awal ini disusun menggunakan *BOW toolkit* dari McCallum dan Kachites [6] untuk mendapatkan frekuensi kata dari tiap-tiap dokumen. Pembobotan dilakukan pada semua frekuensi kata dalam dokumen untuk mencari nilai riil dari masing-masing kata [7]. Aproksimasi atau perkiraan tingkat perwakilan kata dalam dokumen ditunjukkan dengan menghitung persamaan tiap kata. Penelitian ini menggunakan aproksimasi atas dalam merepresentasikan dokumen dengan penambahan proses pengayaan (*enriching*) untuk mencari kata-kata yang berhubungan dengan dokumen, akan tetapi tidak terdapat dalam dokumen itu sendiri [7]. Kata-kata yang berhubungan tersebut dihitung menggunakan teknik *Similarity Roughset Model* (*SRSM*) [3]. Tahap akhir dari sistem klasterisasi dapat menghasilkan beberapa kata kunci yang mewakili dokumen, sehingga dapat menunjukkan hubungan dokumen tersebut dalam kelas yang telah ditentukan.

Penelitian ini bertujuan untuk merepresentasikan sistem klasterisasi teks dengan metode bernama *Max-Max Roughness* (*MMR*) dan menunjukkan perbaikan tingkat kualitas klaster dibandingkan dengan metode *FW-Kmeans* yang dilakukan oleh Jing [4] dengan menggunakan pengukuran *entropy*. Pemilihan *entropy* sebagai parameter kualitas dipilih karena ia dapat menunjukkan tingkat kemurnian dari hasil proses klasterisasi. Kontribusi atau manfaat yang dihasilkan dari klasterisasi teks menggunakan metode *MMR* dengan pengayaan similaritas kata adalah agar didapatkan kualitas klaster yang lebih baik dalam proses klasterisasi teks (klaster dapat menampung data yang bersifat *uncertain*).

## KLASTERISASI TEKS DENGAN METODE *MAX-MAX ROUGHNESS* (*MMR*)

### Pemrosesan Awal (*Preprocessing*)

Pemrosesan awal dokumen sangat penting peranannya dalam memilih kata yang akan dimasukan pada dokumen vektor dan menentukan jumlah kata yang terjadi. Pemrosesan awal sangat menentukan hasil sistem klasterisasi karena menentukan kualitas dan performasi dari data yang mewakili dokumen tersebut. Pemrosesan ini dilakukan dengan menjalankan beberapa prosedur seleksi dokumen, seperti analisa leksikal, penghapusan *header*, *HTML*, *stop word list*, dan *stemming* [6]. Proses dilakukan menggunakan *BOW toolkit* sehingga dihasilkan vektor frekuensi kata terhadap dokumen *VSM*. Vektor ini berukuran $N$ x $M$, dimana $N$ adalah banyaknya dokumen, sedangkan $M$ adalah banyaknya jenis kata yang terdapat dalam data set. Hasil dari *VSM* data model ditunjukkan oleh Gambar 2.

Pada Gambar 2, $d_M$ menunjukkan dokumen dalam *dataset*, sedangkan $t_N$ menunjukkan kata (*term*) yang terdapat dalam dokumen. $w_{MN}$ menunjukkan frekuensi kata dalam dokumen.

### Pembobotan

Pembobotan kata dalam *VSM* pada umumnya menggunakan teori *TF.IDF*. *TF* (*Term Frequency*) adalah banyaknya kata yang muncul dalam sebuah dokumen. Sedangkan *IDF* (*Invers Document Frequency*) adalah pembanding terhadap banyaknya dokumen yang mengandung kata tersebut. *TF.IDF* dihitung dengan menggunakan Persamaan (1).

$$\omega_{ij} = \left(1 + \log t_f\right) * \log \frac{N}{f_D\left(t_i\right)} \quad (1)$$

Dimana :

$t_f$ = Banyaknya kata dalam dokumen.

$N$ = Jumlah dokumen dalam *dataset*.

$f_D(t_i)$ = Jumlah dokumen yang mempunyai kata tersebut.

### Similaritas Kelas

Similaritas kelas atau kelas persamaan merupakan matrik biner berukuran M x M yang disusun dari matrik dokumen. Bila diberikan $U = T = \{t_1, t_2, t_3, \ldots, t_N\}$ dimana $t$ adalah semua kata dalam *dataset*. Hubungan biner $R$ dinyatakan dengan fungsi *Uncertain* $I_\alpha : U \rightarrow 2U :$ seperti dalam Persamaan (2).

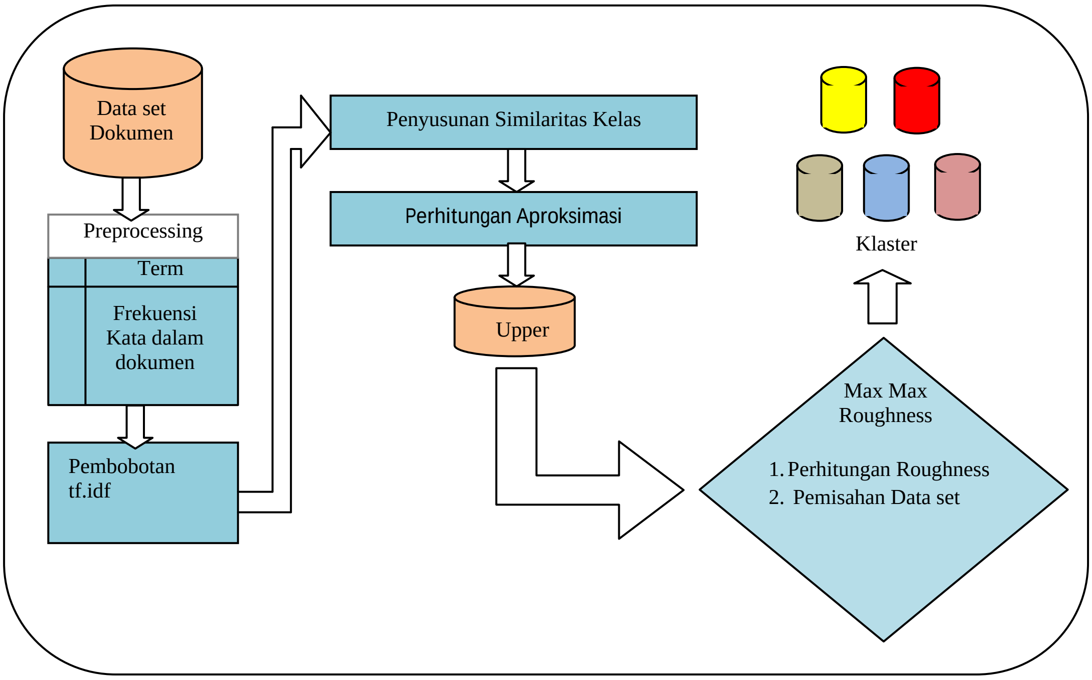
$$I_\alpha(t_i) = \{t_j \in U \mid t_j R t_i\} \quad (2)$$

*Similarity Roughset Model* (*SRSM*) didefinisikan dalam Persamaan (3).

$$I_\alpha(t_i) = \{t_j \mid f_D(t_i, t_j) \geq \alpha \cdot f_D(t_i)\} \cup \{t_i\} \quad (3)$$

$f_D(t_i, t_j)$ adalah banyaknya dokumen yang mengandung kata *i* dan kata *j*.

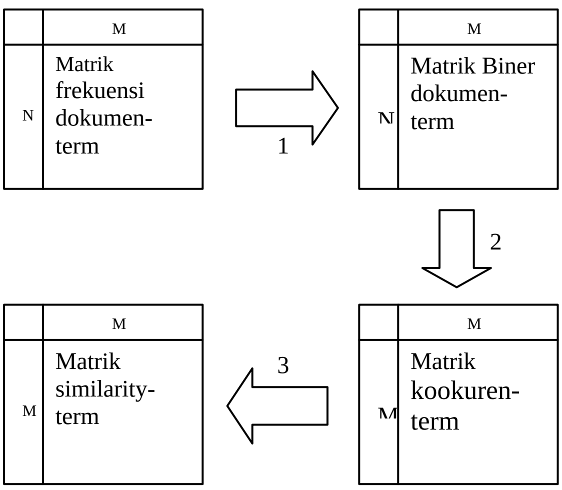
Bila sebuah kata *j* memenuhi Persamaan (3) terhadap kata *i*, maka kata *j* tersebut mempunyai similaritas terhadap kata *i*. Sehingga, elemen matrik dalam similaritas kelas bernilai 1. Similaritas kelas disusun untuk mencari kata yang berhubungan dengan dokumen, tetapi kata tersebut tidak terdapat dalam dokumen itu sendiri.




Gambar 1. Sistem Klasterisasi Teks.

| | $t_1$ | $t_2$ | $t_3$ | … | $t_N$ |
|---|---|---|---|---|---|
| $d_1$ | $w_{11}$ | $w_{12}$ | $w_{13}$ | … | $w_{1N}$ |
| $d_2$ | $w_{21}$ | $w_{22}$ | $w_{23}$ | … | $w_{2N}$ |
| $d_3$ | $w_{11}$ | $w_{11}$ | $w_{11}$ | … | $w_{1N}$ |
| . | . | . | . | . | . |
| . | . | . | . | . | . |
| . | . | . | . | . | . |
| $d_M$ | $w_{M1}$ | $w_{M2}$ | $w_{M3}$ | … | $w_{MN}$ |

Gambar 2. *VSM* Data Model.




Gambar 3. Penyusunan Similaritas Kelas.

Gambar 3 menunjukkan penyusunan similaritas kelas dengan langkah-langkah berikut [7]:

1. Menghitung matrik *binary* dari frekuensi *term* yang terdapat dalam dokumen. Bila *term* tersebut terdapat dalam dokumen, maka diisi dengan 1. Bila tidak terdapat dalam dokumen, maka diisi dengan 0. Matrik ini berukuran *N* x *M*.
2. Menyusun matrik *binary* kookuren. Bila term *i* dan term *j* terdapat dalam sebuah dokumen, maka komponen matrik *i,j* diisi dengan 1. Bila tidak, maka diisi dengan 0. Matrik ini berukuran *M* x *M*.
3. Menyusun matrik similaritas kelas. Matrik similaritas kelas disusun berdasarkan Persamaan (3). Bila memenuhi persamaan tersebut, maka komponen matrik $t_i$ terhadap $t_j$ diisi dengan 1. Bila tidak memenuhi, maka diisi dengan 0.

**Perhitungan Aproksimasi**

Perhitungan aproksimasi disusun berdasarkan Persamaan (3). Aproksimasi yang dipilih dalam penelitian ini aproksimasi atas (*upper aproximation*) karena ia menyertakan unsur pengayaan dalam dokumen. Matrik aproksimasi atas merupakan matrik berukuran *N* x *M*. Pengayaan dilakukan dengan mempertimbangkan similaritas kelas dari kata-kata yang terdapat dalam setiap dokumen. Bila suatu dokumen mempunyai kata *i*, dan tidak mempunyai kata *j*, tetapi kata *j* tersebut memenuhi persamaan similaritas terhadap kata *i*, maka elemen matrik dalam dokumen tersebut bernilai tidak sama dengan nol, walaupun dokumen tersebut tidak mengandung kata *j* tersebut. Bobot dari elemen matrik $d_i, t_j$ didefinisikan dalam Persamaan (4).

$$\min_{t_k \in d_i \omega_{ij}} * \frac{\log \frac{N}{f_D(t_j)}}{1 + \log \frac{N}{f_D(t_j)}} \qquad (4)$$

$\min_{t_k \in d_i \omega_{ij}}$ adalah bobot terkecil dari kata yang dimiliki oleh dokumen $d_i$. Bobot $t_j$ ini tentu saja lebih kecil jika dibandingkan dengan bobot kata yang benar-benar terdapat dalam dokumen. Hal ini dikarenakan kata $t_j$ pada kenyataannya tidak terdapat dalam dokumen $d_i$,, namun mempunyai hubungan similaritas dengan kata yang terdapat dalam $d_i$. Setelah matrik aproksimasi atas terbentuk, dilakukan normalisasi terhadap matrik tersebut. Normalisai dihitung dengan Persamaan (5).

$$\omega_{ij} = \frac{\omega_{ij}}{\sqrt{\sum t_k \in d_i (\omega_{ij})^2}} \qquad (5)$$

## Teori *Roughset*

Pada tiap penelitian penggalian data dilakukan pengklasifikasian informasi yang bertujuan untuk menurunkan dimensi, mengeksplorasi algoritma, dan serta merepresentasikan data. Suatu penelitian dengan nama Teori *Roughset* telah dilakukan untuk mengklasifikasikan informasi yang kurang lengkap [9]. Dalam *Roughset*, *A* adalah satu *set* dari semua atribut yang terdapat dalam obyek *U*. Sedangkan *B* merupakan bagian dari *A* yang bukan himpunan kosong. Obyek $x_i$ dan $x_j$ adalah obyek yang tidak diketahui bila $a(x_i)=a(x_j)$ untuk setiap $a \in B$, sesuai dengan Persamaan (6).

$$\exists\ B \subset A,\ \forall\ a \in B,\ a(x_i) = a(x_j) \qquad (6)$$

Dalam himpunan tersebut, didapatkan batas bawah $\underline{X_B}$ yang merupakan *Union* dari semua *elementary* dalam *X*. Sedangkan batas atasnya adalah *Union* dari *elementary* yang beririsan dengan *X* dan bukan himpunan kosong. Batas atas dan bawah didefinisikan dalam Persamaan (7) dan (8).

$$\underline{X_B} = \{x_i \in U \mid (x_i)_{Ind\,(B)} \subset X\} \qquad (7)$$

$$\overline{X}_B = \{x_i \in U \mid (x_i)_{Ind\,(B)} \cap X \neq 0\} \qquad (8)$$

Dari kedua batas atas dan bawah didapat *Roughness* yang merupakan perbandingan antara aproksimasi bawah dengan aproksimasi atas, sesuai dengan Persamaan (9).

$$R_B(X) = |\underline{X_B}| \,/\, |X_B| \qquad (9)$$

## Teori *Max-Ma x Roughness*

Teori *Roughset* dikembangkan untuk diaplikasikan pada klasterisasi penggalian data [8]. Algoritma *MMR* dihitung dengan mencari *Roughness* maksimal dari perbandingan dua atribut. Bila diberikan sembarang atribut $a_i \in U$, *dimana* $V(a_i)$ merupakan nilai dari atribut $a_i$, maka dapat didefinisikan Persamaan (10).

$$\{R_{aj}(X) \mid a_i = \alpha\} = \frac{|X(a_i = \alpha) a_j|^{lower}}{|X(a_i = \alpha) a_j|^{upper}}$$

$$\text{dengan } a_i, a_i \in A \text{ dan } a_i \neq a_j \qquad (10)$$

Dimana

$X$ = Bagian dari obyek yang memiliki satu buah nilai spesifik.

$\alpha$ = Atribut dari $a_i$.

$R_{a_j}(X)$ = *Roughness* dari *X* mengacu pada attribut $a_i$.

Didapatkan *mean Roughness* dan *MMR* seperti Persamaan (11) yang berkaitan dengan Persamaan (12) dan (13).

$$Rough_{a_j}(a_i) = \frac{\sum_{X=1}^{|V(a_j)|} R_{a_j}(X)}{|V(a_j)|}$$

$$\text{dengan } a_i, a_i \in A \text{ dan } a_i \neq a_j \qquad (11)$$

$$MR(a_i) = \max{}_j \left(Rough_{aj}(a_i)\right)$$

$$\text{dengan } a_i, a_i \in A \text{ dan } a_i \neq a_j \qquad (12)$$

$$MMR(a_i) = \left(\max{}_j \left(Rough_{aj}(a_i)\right)\right)$$

$$\text{dengan } a_i, a_i \in A \text{ dan } a_i \neq a_j \qquad (13)$$

Algoritma *MMR* dibagi menjadi dua prosedur, yaitu prosedur *MMR Main* (Gambar 4) dan prosedur *MMR* (Gambar 5). Gambar 4 menjelaskan perhitungan *MMR* yang diiterasi sampai pemisahan memenuhi klaster.

```
Procedure MMR (U, k)
Begin
    Dataset = U
Do until CNC > k Jika pemisahan belum memenuhi klaster, maka proses
    dilanjutkan
    Call MMRMain (Data set)
    Dataset = Dataset hasil Pemisahan (CNC)
    Loop
End
```

Gambar 4. Algoritma Prosedur *MMR*.

```
Procedure MMRMain (Dataset)
Menghitung roughness untuk semua kata
Menghitung semua pasangan dari a_i
Menghitung Roughness dari kata yang berhubungan dengan a_i
    Menghitung Rough_{a_j}(a_i)   berdasarkan Persamaan  (6)
    Max-Roughness (a_i) Max ( Rough_{a_j}(a_i))  Berdasarkan  Persamaan
(7)
Set Max-Max-Roughness = Max (Max-Roughness (a_i)  Berdasarkan
Persamaan (8)
Menentukan kata pemisahan a_i berdasarkan the Max-Max-Roughness
Menentukan titik pemisahan pada a_i dimana Semua kemungkinan
pemisahan dihitung menggunakan MMR
Lakukan pemisahan berdasarkan elemen biner pada kata yang
dipilih
Data set baru hasil pemisahan terbentu k (CNC)
CNC = CNC + 1
End
```

Gambar 5. Algoritma Prosedur *MMR Main*.

Tabel 1 menunjukkan contoh data yang dipergunakan dalam perhitungan *MMR*. Untuk menyelesaikan perhitungan *MMR* menggunakan *dataset* Tabel 1, maka dilakukan inisialisasi klaster terlebih dahulu. Inisialisasi klaster dilakukan secara subyektif, dimana jumlah klaster ditentukan terlebih dahulu.

Langkah berikutnya adalah menghitung *Roughness* rata-rata dari setiap atribut. *Roughness* rata-rata dari setiap atribut dihitung menggunakan Persamaan (1). Misalkan, suatu atribut *Bobot* memiliki dua jenis atribut, yaitu *Berat* (1,2,3) dan *Ringan* (4,5,6,7,8). Kedua jenis atribut tersebut dihitung rata-rata *Roughness*-nya terhadap semua atribut. Misalkan lagi atribut *Bobot* terhadap *Roda*, maka e*lemen set* dari *Roda* terdiri dari empat jenis, yaitu *Enam* (1,5); *Empat* (2,6); *Tiga* (3,7); dan *Dua* (4,8).

Aproksimasi bawah dari jenis *Berat* di *Bobot* terhadap *Roda* tidak ada yang sama, maka ia bernilai 0. Nilai *Roughness*-nya berarti juga 0. Aproksimasi bawah dari *Ringan* (4,5,6,7,8) terhadap *Roda* adalah 4 dan 8, sedanglan aproksimasi atasnya adalah 1,2,3,4,5,6,7,8. Sehingga, didapatkan *Roughness*-nya 2/8 = 0,25. Rata-rata Roughness adalah 0,125.

Tabel 2 menunjukkan hasil perhitungan *Roughness* rata-rata dari semua atribut pada *dataset* contoh (Tabel 1). Setelah dilakukan perhitungan *Roughness* rata-rata, maka ditentukan pembagian atau partisi *dataset*. Pembagian *dataset* dilakukan dengan melihat hasil perhitungan dari *MMR*. Dari hasil tersebut, didapatkan bahwa atribut *Ukuran* mempunyai *max Roughness* paling besar, sehingga dipilih sebagai atribut yang akan dipartisi. Poin pemisahaan terhadap *dataset* Tabel 1 mempunyai tiga pilihan variasi, yaitu:

1. Pemisahan 1: *Kecil* dan *Sedang-Besar* = (3,7,8) dan (1,2,4,5,6).
2. Pemisahan 2: *Sedang* dan *Kecil-Besar* = (1,6) dan (2,3,4,5,7,8).
3. Pemisahan 3: *Besar* dan *Kecil-Sedang* = (2,4,5) dan (1,3,6,7,8).

Tabel 1. Contoh Data *MMR.*

| Obyek | Bobot | Ukuran | Roda |
|---|---|---|---|
| 1 | Berat | Besar | Enam |
| 2 | Berat | Sedang | Empat |
| 3 | Berat | Kecil | Tiga |
| 4 | Ringan | Besar | Dua |
| 5 | Ringan | Besar | Enam |
| 6 | Ringan | Sedang | Empat |
| 7 | Ringan | Kecil | Tiga |
| 8 | Ringan | Kecil | Dua |

Tabel 2. Contoh Hasil Perhitungan *Roughness.*

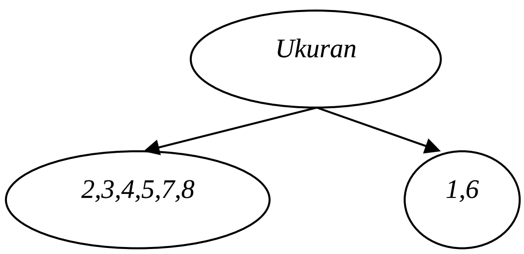
| *Attributes* | *Mean Roughness* | *Max Roughness* |
|---|---|---|
| Bobot | $Rough_{a\ j}$ (Bobot ), j = 2, 3 (0, 0.125) | 0.125 |
| Ukuran | $Rough_{aj}$ (Ukuran ), j = 1, 3 (0, 0.667) | Max: 0.667 |
| Roda | $Rough_{a}$ (Roda ) , j = 1, 2 (0, 0.250) | 0.250 |

Tabel 3. Hasil Contoh Perhitungan *Max Roughness*.

| Pembagian | *Mean Roughness* | *Max Roughness* |
|---|---|---|
| Kecil & Sedang-Besar | Rough (Ukuran), j=1,3 (0;0,583) | 0,583 |
| Sedang & Kecil-Besar | Rough (Ukuran), j=1,3 (0;1) | Max = 1 |
| Besar & Kecil-Sedang | Rough (Ukuran), j=1,3 (0;0,583) | 0,583 |

Tabel 4. *Matrix Confusion* Contoh Klasterisasi.

| Pemisahan | Klaster 1 | Klaster 2 |
|---|---|---|
| Kelas 1 | 100 | 0 |
| Kelas 2 | 7 | 93 |




Gambar 6. Pemisahan *Dataset*.

Untuk setiap kemungkinan dilakukan perhitungan *MMR* ulang, sehingga didapatkan variasi pemisahan dengan *Roughness* paling besar ditunjukkan oleh variasi *Sedang* dan *Kecil-Besar* seperti ditunjukkan pada Tabel 3.

Gambar 6 menunjukkan pemisahan *dataset* berdasarkan perhitungan *Roughness* rata-rata dari atribut ukuran setelah dilakukan variasi penggabungan data. Jika *dataset* telah dipisahkan, maka Algoritma *MMR* dapat diulang untuk *dataset* hasil pemisahan yang mempunyai kapasitas paling besar sampai didapatkan jumlah klaster yang diinginkan.

### *Entropy*

Jika *dataset* telah dikelompokkan ke dalam masing-masing klaster, maka dilakukan pengujian *entropy* untuk menunjukkan kualitas dari klaster yang terbentuk. *Entropy* adalah suatu parameter yang menunjukkan tingkat kemurnian dari klaster. *Entropy* dihitung berdasarkan *matrix confusion* hasil klasterisasi dalam Persamaan (14).

$$Entropy = \sum_{l=1}^{k} \frac{n_l}{n} \left( -\frac{1}{\log k} \sum_{h=1}^{k} \frac{n_{h,l}}{n_l} . \log \frac{n_{h,l}}{n_l} \right) \quad (14)$$

Dimana :

$n_h$, $n_l$ = Jumlah dokumen yang terdapat dalam kelas $C_h$ dan klaster $S_l$.

$n$ = Jumlah total dokumen dokumen.

$n_{h,l}$ = Jumlah dokumen yang terdapat dalam keduanya ($C_h$ dan $S_l$).

Tabel 4 menunjukkan contoh hasil klasterisasi dengan *dataset* sebanyak 200 dokumen. Pada klaster 1, didapatkan *dataset* dengan jumlah 107 dikelompokkan menjadi satu klaster, dengan distribusi kelas 1 sebanyak 100 dokumen, dan kelas 2 sebanyak tujuh dokumen. Dalam klaster 1 ini, terdapat penyimpangan tujuh buah dokumen yang bukan satu kelas dikelompokkan ke dalam klaster yang sama. Dengan menggunakan Persamaan (15), didapatkan *entropi*= 0,1830. Keakuratan hasil *entropi* berada pada jangkauan 0 - 1 dimana semakin kecil hasil *entropi*-nya, maka kualitas klaster semakin baik.

## HASIL DAN PEMBAHASAN

Uji coba pada metode *MMR* dilakukan terhadap empat *dataset* yang terdiri dari 20 *Newsgroups* (http://people.csail.mit.edu/jrennie/20Newsgroups). Tabel 5 menunjukkan *dataset* yang disusun dari 20 kelompok berita. Masing-masing *dataset* disusun sesuai dengan skenario dalam uji coba untuk memenuhi kualitas hasil yang diharapkan. Kelas menunjukkan kelompok berita yang dipilih dalam pengujian, sedangkan $n_d$ menunjukkan jumlah data yang dipilih secara *random* dari setiap kelas. Skenario pemilihan *dataset* terdiri dari empat skenario, yaitu *A2*, *A4-U*, *B2*, dan *B4-U*.

*Dataset A2* (skenario 1) dipilih karena terdiri dari dua jenis kelas dengan topik yang sangat berbeda, yaitu topik *atheisme* dan komputer. *Dataset A*4-U (skenario 2) disusun berdasarkan jenis kelas yang mempunyai perbedaan topik sangat jelas dan distribusi jumlah dokumen yang bervariasi. Skenario 2 mempunyai distribusi 120, 100, 59, dan 20. Sedangkan *dataset B*2 (skenario 3) terdiri dari dua jenis kelas dengan topik yang sama yaitu tentang politik. *B*4-U (skenario 4) mempunyai jenis kelas yang hampir sama topiknya yaitu komputer, dengan distribusi dokumen yang bervariasi.

Dari hasil uji coba didapatkan kata kunci yang mewakili dokumen dalam klaster. Kata-kata kunci tersebut menunjukkan kata yang mempunyai *Roughness* paling tinggi dalam dokumen. Masing-masing kata kunci dan frekuensinya dari hasil uji coba skenario 1 sampai dengan 4 ditunjukkan oleh Tabel 6.

Pengujian kualitas dengan perhitungan *entropy* dari hasil klasterisasi dengan menggunakan metode *MMR* terhadap masing masing *dataset* ditunjukkan oleh Tabel 7. Dari Tabel 7 didapatkan perbandingan hasil pengujian *entropy* menggunakan *MMR* dan *FW-KMeans*. Hasil pengujian pada *A2* menunjukkan bahwa kualitas hasil pada *MMR* memiliki nilai lebih baik 0,0227 dari pada *FW-Kmeans*. Sedangkan pada *A4-U* menunjukkan bahwa kualitas hasil pada *MMR* memiliki nilai lebih baik 0,330. Hasil pengujian *B2* menunjukkan bahwa kualitas hasil pada *MMR* memiliki nilai lebih baik 0,2091 dari pada *FW-Kmeans* dan pada *B4-U* perbaikan yang didapatkan adalah 0,0682.

Tabel 5. *DataSet* Uji Coba 20 *Newsgroup*.

| *Dataset* | Kelas | $n_d$ | *Dataset* | Kelas | $n_d$ |
|---|---|---|---|---|---|
| *A2* | *Alt.atheism* | 100 | *B2* | *Talk.politics.mideast* | 100 |
| | *Comp.graphics* | 100 | | *Talk.politic.misc* | 100 |
| *A4-U* | *Comp.graphics* | 120 | *B4-U* | *Comp.graphics* | 120 |
| | *Rec.sport.baseball* | 100 | | *Comp.os.ms-windows* | 100 |
| | *Sci.space* | 59 | | *Rec.autos* | 59 |
| | *Talk.politics.mideast* | 20 | | *Sci electronics* | 20 |

Tabel 6. Kata Kunci Klaster.

| *A2* | | *A4-U* | | *B2* | | *B4-U* | |
|---|---|---|---|---|---|---|---|
| Kata | *f* | Kata | *f* | Kata | *f* | Kata | *f* |
| *writes* | 154 | *Space* | 220 | *israel* | 176 | *image* | 231 |
| *article* | 145 | *season* | 172 | *armenians* | 124 | *windows* | 229 |
| *file* | 140 | *people* | 172 | *israeli* | 104 | *graphics* | 188 |
| *graphics* | 110 | *israel* | 123 | *armenian* | 103 | *File* | 162 |
| *people* | 110 | *graphics* | 111 | *turkish* | 70 | *program* | 105 |
| *format* | 105 | *nasa* | 90 | *jewish* | 67 | *images* | 102 |
| *god* | 87 | *israeli* | 76 | *arab* | 57 | *Os* | 100 |
| *claim* | 42 | *jewish* | 64 | *armenia* | 57 | *dos* | 96 |
| *evidence* | 33 | *game* | 63 | *peace* | 54 | *Car* | 71 |
| *atheists* | 21 | *arab* | 55 | *turkey* | 52 | *Pc* | 60 |

Tabel 7. Perbandingan *Entropi MMR* terhadap *FW-Kmeans*.

| Metode | *A*2 | *A*4-U | *B*2 | *B*4-U |
|---|---|---|---|---|
| *FW-KMeans* | 0,2057 | 0,1513 | 0,4014 | 0,2314 |
| *MMR* | 0,1830 | 0,1083 | 0,1919 | 0,1632 |

Dari hasil uji coba klasterisasi menggunakan *MMR*, secara keseluruhan didapatkan bahwa klasterisasi *MMR* mempunyai tingkat kualitas rata-rata yang lebih baik dari klasterisasi dengan menggunakan metode *FW-Kmeans* [4]. Tingkat perbaikan yang dihasilkan adalah sebesar 30,28% walaupun terjadi penurunan kualitas pada akhir proses pemisahan dokumen dalam *dataset*. Gejala tersebut ditunjukkan dengan adanya kata yang dikelompokkan ke dalam kelas yang berbeda. Hal ini terjadi dikarenakan:

1. Metode klasterisasi *MMR* melakukan langkah klasterisasi dengan mencari *Roughness* atau derajat kecenderungan dari masing-masing kata atau kata yang mewakili dokumen secara parsial terhadap masing-masing kelas. Sebagian besar dari kata-kata dalam dokumen tersebut mempunyai tingkat atau bobot kecenderungan yang sangat kecil terhadap kelas yang sudah ditentukan. Pada langkah pemisahan di awal proses, kata-kata dalam dokumen dihitung *Roughness*-nya terlebih dahulu, kemudian dokumen dipisahkan sesuai dengan *Roughness* sehingga dapat dengan akurat menentukan bahwa dokumen yang memiliki kata tersebut masuk ke dalam kelas tertentu. Pada pemisahan akhir, terjadi pemisahan kata yang tidak sesuai dengan kelasnya karena kata-kata tersebut mempunyai bobot atau tingkat perwakilan yang sangat kecil terhadap kelas.
2. Metode *MMR* menggunakan variabel kata secara parsial dan memisahkan data set berdasarkan kata yang mempunyai *Roughness* paling besar. Sehingga kata-kata lain yang terdapat dalam dokumen yang sama tidak mempengaruhi pemisahan.

## SIMPULAN DAN SARAN

Dari penelitian didapatkan simpulan sebagai berikut:

1. Metode *MMR* merupakan klasterisasi teks model baru yang digunakan untuk membentuk klaster berdasarkan kata yang merupakan perwakilan dari kelas. Metode ini bersifat antar obyek dengan perbandingan *Roughness* dari kata sebagai atribut dokumen sehingga dapat mengatasi permasalahan *uncertainty* pada sistem klasterisasi.
2. Klasterisasi dengan metode *MMR* dapat menghasilkan kualitas klaster yang lebih baik dibandingkan dengan metode *FW-Kmeans*, dengan tingkat perbaikan rata-rata sebesar 30,28%.

Saran yang dapat diberikan untuk penelitian selanjutnya adalah metode *MMR* digunakan untuk menentukan kelas baru. Hal ini dapat dilakukan berdasarkan kata-kata yang mempunyai *Roughness* tinggi tetapi tidak mewakili atau tidak berhubungan dengan kelas yang sudah ada. Pengukuran kualitas dengan parameter lain seperti akurasi, *F-Score*, dan *NMI* juga dimungkinkan dalam penelitian selanjutnya.

## DAFTAR PUSTAKA

[1] Remesh KM. *Application of Roughset in Data Mining*. Thesis. Kerala: Department of Computer Science Cochin India University. 2008.

[2] Ho TB and Kawasaki S. *Text mining with Tolerance Rough Set Models*. Ishikawa: Japan Advanced Institute of Technology. 2007.

[3] Thanh NC, Koichi Y, and Muneyuki U, *Evaluation of Document Clustering based on Similarity Rough Set Model*. Thesis. Nagaoka: Nagaoka University of Technology Japan. 2006.

[4] Jing LP, Michael KN, Yang XH, and Huang JZ. A Text Clustering System based on K-means Type Subspace Clustering and Ontology. *International Journal of Intelligent Technology*. 1: 91-103. 2006.

[5] Parmar D and Wu T. *A Clustering Algorithm for Supplier Base Management*. Berlin: Springer Berlin. 2006.

[6] McCallum R and Kachites A. *Bow: A Toolkit for Statistical Language Modeling, Text Retrieval, Classification, and Clustering.* 1998. URL: http://www.cs.cmu.edu/~mccallum/bow, diakses tanggal 23 Mei 2010.

[7] Lang NC. A Tolerance Roughset Approach to Clustering web Search Result. *Proceedings of 8th European Conference on Principles and Practice of Knowledge Discovery in Database*. 515-517. 2003.

[8] Mazlack L, He A, Zhu Y, and Coppock S., 2000, A Rough Set Approach in Choosing Partitioning Attributes. *Proceedings of the ISCA 13th International Conference (CAINE-2000).* 1-6. 2000.

[9] Pawlak Z. Rough sets. *International Journal of Computer and Information Sciences*. 11: 341-356. 1982.