Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)
Edisi. .. Volume. .., Bulan 20.. ISSN : 2089-9033

SVM UNTUK COREFERENCE RESOLUTION BAHASA INDONESIA YANG MENGANDUNG ENTITAS JAMAK

Syifa Muhammad Husni[1], Ken Kinanti Purnamasari[2]

[1,2] Teknik Informatika – Universitas Komputer Indonesia
Jl. Dipatiukur 112-114 Bandung
Email : syifamh94@gmail.com[1], ken.kinanti@email.unikom.ac.id[2]

ABSTRAK

Coreference Resolution berfungsi untuk menemukan dua ungkapan *natural language* merujuk pada entitas yang sama. Pada penelitian sebelumnya, *Coreference Resolution* digunakan untuk mengidentifikasi kata ganti tunggal dan entitas nama seseorang. Data masukan yang digunakan adalah teks novel berbahasa Indonesia. Dalam sebuah novel akan mengandung rangkaian cerita kehidupan seseorang dan orang-orang di sekelilingnya. Sehingga kata ganti untuk orang bukan hanya kata ganti tunggal, namun juga terdapat banyak orang yang sedang diceritakan atau kata ganti jamak. Sehingga pada penelitian ini akan dibuat *Coreference Resolution* dengan kata ganti jamak dan kata ganti pada teks berbahasa Indonesia dengan menggunakan metode *Support Vector Machine* (SVM) agar dapat dapat mengukur hasil akurasi yang dihasilkan ketika kata jamak teridentifikasi. Adapun data masukan yang akan diolah, akan melalui tahap *preprocessing* hingga ekstraksi fitur untuk mendapatkan format SVM yang akan digunakan pada metode SVM, Pada penelitian ini terdapat 7 fitur yang digunakan untuk mendeteksi *Coreference Resolution* diantaranya *i-pronoun*, *j-pronoun*, *i-propernoun*, *j-propernoun*, *distance of string*, *distance of word* dan *string match*. Berdasarkan hasil pengujian, nilai rata-rata akurasi yang didapat pada penelitian ini adalah sebesar 61.41%.

Kata kunci : *Coreference Resolution*, *Suppport Vector Machine*, *Natural Language Processing*, POSTAG, *Text Mining*.

1. PENDAHULUAN

Coreference Resolution adalah proses untuk mengidentifikasi kesetaraan antar entitas dan berfungsi untuk menemukan dua ungkapan bahasa alami yang merujuk pada entitas yang sama [1]. *Coreference Resolution* telah diteliti dalam berbagai bahasa termasuk bahasa Indonesia. Pada penelitian sebelumnya, *Coreference Resolution* untuk bahasa Indonesia telah dibangun, yaitu untuk mengidentifikasi tokoh yang sedang diceritakan pada teks bahasa Indonesia, kata yang dapat teridentifikasi hanya untuk kata ganti tunggal dan entitas nama seseorang. Dalam penelitiannya, yang menjadi data masukan adalah novel bahasa Indonesia. Novel adalah karangan prosa yang panjang dan mengandung rangkaian cerita seseorang dengan sekelilingnya yang menonjolkan watak dan sifat setiap perilaku [2].

Pada penelitian sebelumnya, metode *Support Vector Machine* (SVM) digunakan untuk kasus novel bahasa Indonesia. Dalam penelitian tersebut digunakan 2 data latih berupa novel dengan total jumlah pasang sejumlah 2788 pasangan coreference dan data uji berupa 5 kutipan novel yang mengandung kata ganti atau entitas bernama, sehingga menghasilkan rata-rata akurasi sebesar 50.14% dengan akurasi tertinggi 63% dan akurasi terendah 38.8% [3].

Berdasarkan uraian di atas, nilai akurasi pada *Coreference Resolution* dengan menggunakan kata ganti jamak belum diketahui, sehingga pada penelitian ini akan melakukan penambahan pendeteksian pada kata ganti jamak dengan menggunakan data latih dan uji dari novel berbahasa Indonesia. Metode yang akan digunakan untuk klasifikasi adalah metode *Support Vector Machine* (SVM).

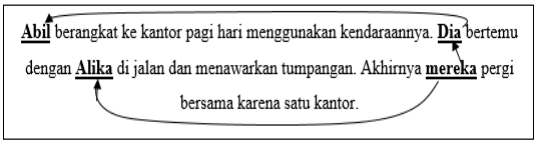
Penelitian ini bermaksud untuk mengembangkan sistem pendeteksi koreferensi yang tepat pada sebuah kalimat. Adapun tujuan yang ingin dicapai dari penelitian ini adalah untuk mengetahui nilai akurasi pada *Coreference Resolution* dengan menyediakan pendeteksi untuk kata benda jamak menggunakan metode SVM.

2. ISI PENELITIAN

2.1 *Coreference Resolution*

Coreference Resolution merupakan subtugas dari *Natural Language Processing* (NLP), dan juga merupakan proses untuk mengidentifikasi dua ungkapan bahasa alami yang mengacu pada

entitas yang sama. Tujuan *Coreference Resolution* adalah untuk mengidentifikasi kesetaraan antar entitas maupun antara pronomina dan entitas yang dikenali dalam fase pengenalan entitas bernama [1]. Berikut adalah ilustrasi *Coreference Resolution* dalam penelitian ini.



Gambar 1. Contoh *Coreference Resolution*

2.2 *Preprocessing*

Dalam pembangunan sebuah sistem *Coreference Resolution*, perlu dilakukan tahapan yang perlu dilakukan supaya data masukan dapat diproses oleh sistem. Tahap yang perlu dilakukan adalah tahap untuk mengolah data yang belum terstruktur supaya menjadi terstruktur sesuai dengan kebutuhan data sebelum diolah pada tahap berikutnya atau yang biasa disebut dengan *preprocessing*. Dalam penelitian ini *preprocessing* yang dilakukan adalah *Case Folding*, *Filtering*, *Tokenizing*, dan POS *Tagging*.

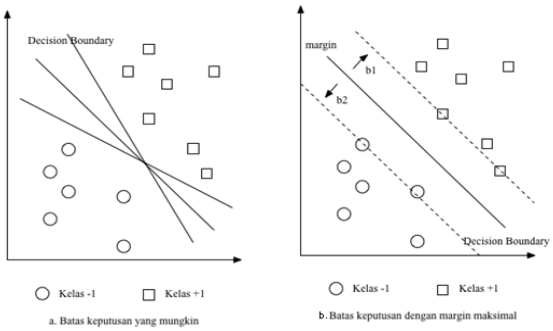
2.3 Ekstraksi Fitur

Pada penelitian ini data yang telah diubah sesuai dengan kebutuhan data sistem akan di ektrak menggunakan beberapa fitur [1]. Dimana ekstraksi digunakan untuk mengambil informasi yang penting dari suatu data untuk digunakan pada tahap klasifikasi SVM. Sistem ini menggunakan beberapa fitur yang berguna untuk menentukan apakah dua ungkapan *coreference* atau tidak, fitur yang digunakan yaitu.

a. Fitur untuk mendeskripsikan kata-*i(i-Pronoun dan i-Propernoun).*
b. Fitur untuk mendeskripsikan kata-*j(j-Prononun dan j-Propernoun).*
c. *Fitur untuk relasi antar kata (Distance Of Word, Distance Of Sentence, dan String Match).*

2.4 *Support Vector Machine*

Support Vector Machine (SVM) merupakan salah satu algoritma *Machine Learning* yang paling populer untuk klasifikasi dan regresi. Konsep SVM secara sederhana dapat dijelaskan sebagai usaha untuk mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua buah kelas pada input *space*, *hyperplane* pemisah dapat ditemukan dengan mengukur magin, margin merupakan jarak terdekat antara *hyperplane* dengan data dari masing-masing kelas. Subset data training yang paling dekat disebut dengan *support vector*[4]. Berikut adalah ilustrasi dari SVM.



Gambar 2. Batas Keputusan Data Set

SVM digunakan untuk mencari fungsi pemisah (klasifier) atau *hyperplane* optimal yang dapat memisahkan dua set data. Untuk mencari fungsi pemisah tersebut yaitu dengan memaksimalkan margin atau jarak antara dua set objek dari kelas yang berbeda dan dua set objek tersebut adalah *support vector*. Fungsi *linear* merupakan pemisah dalam linear SVM. Data latih akan dinyatakan oleh $(x_i y_i)$, fitur set untuk data latih kelas ke-*i* dinyatakan oleh $x_i = \{x_1, x_2, \dots, x_{iq}\}$, dan label kelas dinyatakan oleh $y_i \in \{-1, 1\}$. *Hyperplane* pemisah didefinisikan dengan persamaan berikut.

$$wx_i + b = 0 \quad (1)$$

Data x_i terbagi ke dalam dua kelas. Data yang termasuk kelas -1 (sampel negatif) didefinisikan sebagai vektor yang memenuhi pertidaksamaan di bawah ini.

$$wx_i + b < 0 \; untuk \; y_i = -1 \quad (2)$$

Sedangkan data yang termasuk kelas +1 (sampel positif) didefinisikan sebagai fektor yang memenuhi pertidaksamaan di bawah ini.

$$wx_i + b > 0 \; untuk \; y_i = +1 \quad (3)$$

Dimana:
x_i = data *input*
y_i = label yang diberikan
w = nilai dari bidang normal
b = posisi bidang relative terhadap pusat koordinat

Parameter w dan b adalah parameter yang akan dicari nilainya. Apabila label data y_i = -1, maka persamaan berikut ini akan menjadi pembatasnya.

$$wx_i + b \leq -1 \quad (4)$$

Sedangkan apabila label data yi = +1, maka pembatas menjadi persamaan di bawah ini.

$$wx_i + b \geq +1 \quad (5)$$

Nilai x_i dan y_i dapat dicari ketika nilai tiap pasangan dari hasil ekstraksi fitur dan inisialisasi kelas telah didapatkan. Hasil dari ekstraksi fitur diubah ke dalam bentuk format

data SVM (x), sedangkan data kelas menjadi label data SVM (y).

Langkah pertama yaitu cara mendapatkan nilai α_i adalah mengubah setiap data *input* menjadi nilai vektor (*support vector*) = (*y x*). Kemudian nilai vektor dari setiap data *input* dimasukkan ke dalam persamaan kernel *trick phi* $(\emptyset)$ berikut.

$$\phi\begin{bmatrix}p_i\\v_i\end{bmatrix} = \begin{cases}\sqrt{{p_n}^2+{v_n}^2} > 2, maka \begin{bmatrix}\sqrt{{p_n}^2+{v_n}^2}-p_i+|p_i-v_i|\\ \sqrt{{p_n}^2+{v_n}^2}-v_i+|p_i-v_i|\end{bmatrix}\\ \sqrt{{p_n}^2+{v_n}^2} \le 2, maka \begin{bmatrix}p_i\\v\end{bmatrix}\end{cases} \quad (6)$$

Nilai *p* dapat diperoleh dari persamaan *kernel linear* untuk nilai *x* berikut.

$$p = \sum_{i=1}^{n} {x_i}^T x_j \ (i,j = 1,\dots,n) \quad (7)$$

Nilai *v* dapat diperoleh dari persamaan *kernel linear* untuk nilai y berikut.

$$v = \sum_{i=1}^{n} {y_i}^T y_j \ (i,j = 1,\dots,n) \quad (8)$$

Hasil perhitungan dari substitusi nilai x dan nilai y ke persamaan 6 diberi nilai bias = 1 untuk mendapatkan jarak tegak lurus yang optimal dengan mempertimbangkan vektor positif. Kemudian cari parameter α_i, dengan terlebih dahulu mencari nilai fungsi setiap abstrak menggunakan persamaan 9. Dengan memperhatikan *i*,=1,..,*n,* nilai α_i dapat dicari pada persamaan linear menggunakan persamaan 10 berikut.

$$\sum_{i=1,j=1}^{n} \alpha_i {S_i}^T S_j \quad (9)$$

$$\sum_{i=1,j=1}^{n} \alpha_i {S_i}^T S_j = v_i \quad (10)$$

Setelah parameter α_i didapatkan, kemudian masukkan ke persamaan berikut.

$$\hat{W} = \sum_{i=1,j=1}^{n} \alpha_i S_i \quad (11)$$

Hasil yang didapatkan dari persamaan 11, selanjutnya digunakan ke dalam persamaan 12 untuk mendapatkan nilai *w* dan *b.*

$$y = wx + b \quad (12)$$

Sedemikian sehingga didapatkanlah nilai *w* dan *b* (nilai *hyperplane*) untuk mengklasifikasikan kedua kelas.

Jika sebuah fungsi dapat memenuhi Teorema Mercer, yang menyatakan bahwa matriks kernel yang dihasilkan harus bersifat semi *positive* semi *definite*, maka fungsi tersebut dapat dikatakan sebagai fungsi kernel [5]. Dalam penelitian ini kernel yang digunakan adalah Kernel Linear.

$$K(x_i, x) = {x_i}^T x \quad (13)$$

Proses klasifikasi metode machine yang digunakan dalam sistem ini adalah LIBSVM, yaitu *library* yang dibuat oleh Garbriel Couto.

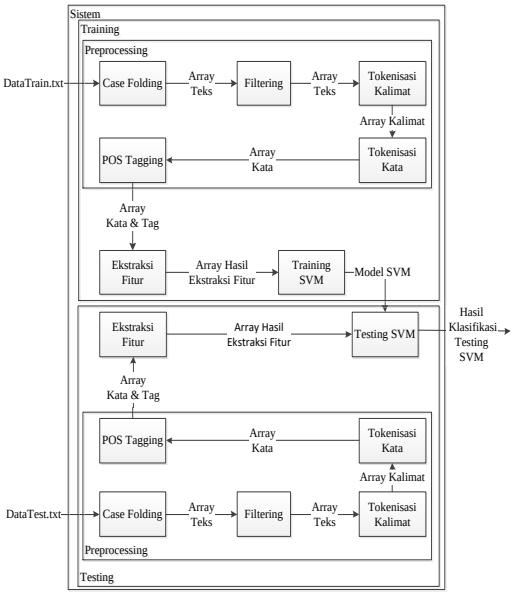
2.5 Analisis Masalah

Berdasarkan pengamatan kembali terhadap novel yang digunakan pada penelitian sebelumnya, banyak ditemukan kata ganti jamak yang belum dapat terdeteksi oleh sistem. Sedangkan kata jamak sendiri termasuk ke dalam jenis kata ganti (*personal pronouns*). Data latih yang digunakan pada penelitian sebelumnya hanya melakukan pendeteksian terhadap kata ganti tunggal dan nama orang, sehingga teks yang memiliki kata ganti jamak tidak digunakan saat pengujian. Hal tersebut menjadikan *Coreference Resolution* sebelumnya menjadi terbatas dalam mendeteksi, karena untuk kata ganti jamak akan terdeteksi sebagai jenis kata ganti *Personal Pronouns* (PRP). Oleh karena itu, penelitian ini akan memanfaatkan fitur i-pronoun, i-propernoun, j-pronoun, dan j-propernoun untuk mengetahui jenis kata dan fitur distance of word (DOW), distance of sentence (DOS), dan string match untuk relasi antara kata yang akan dibandingkan.

2.6 Analisis Solusi

Berdasarkan penjelasan masalah yang sudah dijelaskan sebelumnya, dengan dibuatnya sistem *Coreference Resolution*, Dalam membangun sistem, akan ada beberapa tahap yang akan dilakukan yaitu tahap training untuk data latih, tahap testing untuk data testing, tahap klasifikasi menggunakan SVM dan perhitungan akurasi *Coreference Resolution*.

Berikut adalah gambaran umum dari sistem *Corefence Resolution* yang dibangun.



Gambar 3. Gambaran Umum Sistem

2.6 Analisis Data Masuk

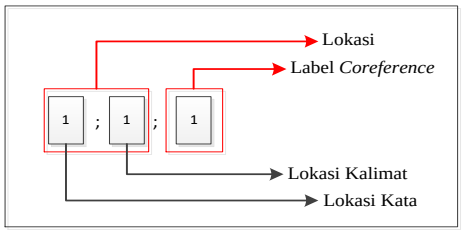
Data masukan yang akan diproses oleh sistem adalah data *training* dan data testing, pada penelitian sebelumnya, dilakukan pemberian label terhadap data *training* untuk menyatakan *coreference* yang akan digunakan pada tahap klasifikasi. Berikut adalah contoh dari potongan teks novel yang berjudul "Surga Sungsang" yang akan digunakan sebagai data training.

Tabel 1. Teks Masukan Data *Training*

Dan yang lebih penting **ia** tak ingin ikan piaraannya kepanasan dan **Zaenab** hangus terbakar. "**Kita** hanya butuh hujan. **Kita** harus memohon Allah agar memberikan sihir hujan!" **Zaenab** mendesis lagi. "Hujan? Untuk apa?" tanya **Kufah** tak mengerti maksud **Zaenab**.

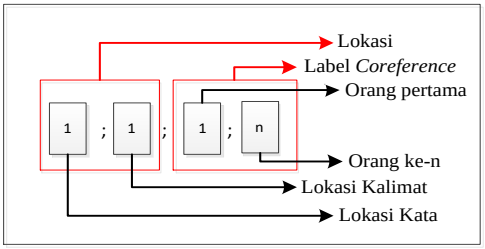
Dari potongan teks novel di atas, terdapat tujuh kata ganti yang terdiri dari empat kata ganti orang (Zaenab, Zaenab, Kufah, Zaenab) dan tiga kata ganti tunggal (ia, Kita, Kita).

Untuk dapat mendeteksi kata-kata tersebut, maka harus diberikan label sebagai kunci untuk pengklasifikasian. Label yang diberikan terdiri dari tiga angka yang dipisahkan oleh titik koma (;) untuk kata ganti tunggal dan empat angka atau lebih untuk kata ganti jamak yang dipisahkan oleh titik koma (;). Contoh pada gambar berikut.



Gambar 4. Data Label Tunggal [3]

Format data label tunggal digunakan untuk memberikan tanda kepada kata ganti tunggal ataupun nama seperti saya, dia, ia, kamu, Zaenab, Kufah.



Gambar 5. Data Label Jamak

Format data label jamak digunakan untuk memberikan tanda kepada kata ganti jamak seperti kata mereka, kita, kami, kalian. Untuk mendapatkan lokasi kata dan kalimat maka dapat dilihat pada tabel berikut.

Tabel 2. Lokasi Kalimat dan Kata

No	Teks Novel
1	Dan yang lebih penting **ia** tak ingin ikan 1 2 3 4 **5** 6 7 8 piaraannya kepanasan dan **Zaenab** hangus 9 10 11 12 13 terbakar. 14
2	"**Kita** hanya butuh hujan. **1** 2 3 4
3	**Kita** harus memohon Allah agar **1** 2 3 4 5 memberikan sihir hujan!" **Zaenab** 6 7 8 **9** mendesis lagi. 10 11
4	"Hujan? Untuk apa?" tanya **Kufah** tak 1 2 3 4 **5** 6 mengerti maksud **Zaenab**. 7 8 **9**

Untuk pemberian label *coreference* yaitu berdasarkan kata ganti orang yang pertama muncul tanpa mempunyai rujukan *coreference.* Sehingga apabila muncul kata ganti orang pertama akan diberikan label *coreference*-nya adalah 1, apabila muncul kata ganti berikutnya dengan orang yang sama maka tetap 1, dan apabila berbeda maka menjadi 2 dan sampai seterusnya tergantung seberapa banyak tokoh yang akan muncul. berikut label *coreference* untuk data masukan di atas.

Tabel 3. Label Coreference

No	Kata	Label *Coreference*
1	Ia	1
2	Zaenab	2
3	Kufah	3
4	Kita	1, 2

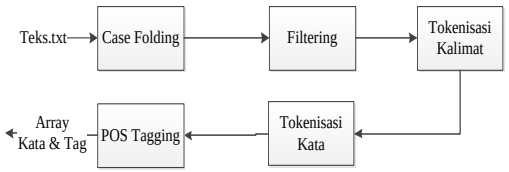
Format data label tunggal dan jamak dapat dilihat pada tabel berikut, dimana untuk lokasi kata dan kalimat dapat dilihat pada tabel 2.

Tabel 4. Data Label

No	Kata	Lokasi		Label *Coreference*	Data Label
		Kata	Kalimat		
1.	ia	5	1	1	5;1;1
2.	Zaenab	12	1	2	12;1;2
3.	Kita	1	2	1, 2	1;2;1;2
4.	Kita	1	3	1, 2	1;3;1;2
5.	Zaenab	9	3	2	9;3;2
6.	Kufah	5	4	2	5;4;3
7.	Zaenab	9	4	3	9;4;2

2.7 *Preprocessing*

Preprocessing harus dilakukan terhadap data masukan yang akan diolah, agar mendapatkan hasil yang lebih maksimal dari setiap tahap ke tahap selanjutnya. Tahap *preprocessing* yang akan dilakukan yaitu case *folding*, *filtering*, *tokenizing* kalimat, *tekenizing* kata, dan POS *tagging*. Tahap preprocessing dapat di lihat pada gambar berikut.



Gambar 6. *Preprocessing*

2.8 Ektraksi Fitur

Ekstraksi fitur adalah tahap untuk mendapatkan data informasi yang dibutuhkan dari data sebelumnya. Kata kunci yang digunakan untuk ekstraksi fitur ini adalah kata ganti orang seperti nama, tunggal, dan jamak. Pada penelitian ini kata jamak ditambahkan untuk menambah informasi kata kunci. Karena kata jamak termasuk ke dalam kata ganti, berikut adalah 7 fitur yang akan digunakan untuk proses klasifikasi menggunakan metode SVM adalah sebagai berikut.

Tabel 5. Ekstraksi Fitur

Fitur untuk mendeskripsikan *kata-i* .	
i-Pronoun	mendeskripsikan nilai *i-pron* yang dideteksi POS tag sebagai PRP. Jika benar maka nilainya 1, jika tidak nilainya 0.
i-Propernoun Pharse	mendeskripsikan nilai *i-prop* yang dideteksi POS tag sebagai NNP. Jika benar maka nilainya 1, jika tidak nilainya 0.
Fitur untuk mendeskripsikan *kata-j*	
j-Pronoun	mendeskripsikan nilai *j-pron* yang dideteksi POS tag sebagai PRP. Jika benar maka nilainya 1, jika tidak nilainya 0.
j-Propernoun Pharse	mendeskripsikan nilai *j-prop* yang dideteksi POS tag sebagai NNP. Jika benar maka nilainya 1, jika tidak nilainya 0.
Fitur untuk relasi antara *kata-i* dan *kata-j*	
Distance Of Words (DOW)	mendeskripsikan jarak antar kata *i* dan *j*, nilai ini akan didapat dari jarak kata *j* dikurangi jarak kata *i* lalu dibagi dengan jumlah kata pada dokumen teks tersebut.
Distance Of Sentence (DOS)	medeskripsikan jarak antar kalimat *i* dan *j*, nilai ini akan didapat dari jarak kalimat *j* dikurangi jarak kalimat *i* lalu dibagi dengan jumlah kata pada dokumen teks tersebut.
String Match	mendeskripsikan string pada *i* dan *j*, jika string pada *i* dan *j* sama maka nilainya 1, jika berbeda maka nilainya 0.

Ekstraksi fitur terbagi menjadi dua, ekstraksi fitur untuk data training dan data testing, berikut ini adalah contoh untuk data masukan untuk *training*. Berikut adalah contoh masukan untuk data *training*.

Tabel 6. Contoh Data Masukan Ekstraksi Fitur

Data Masukan
dan yang lebih penting **ia** tak ingin ikan piaraannya kepanasan dan **zaenab** hangus terbakar. **kita** hanya butuh hujan. **kita** harus memohon allah agar memberikan sihir hujan **zaenab** mendesis lagi. hujan untuk apa tanya **kufah** tak mengerti maksud **zaenab**.

Pada tahap ekstraksi fitur untuk data training, label.txt dimanfaatkan untuk menandakan kata yang akan dideteksi sehingga dapat membuat pasangan kata-*i* dan kata-*j*. Dalam menentukan pasangan kata-*i* dan kata-*j*, lokasi kata-*i* > kata-*j* (i>j). Dimana lokasi kata yang dimaksud adalah jumlah keseluruhan kata pada teks novel yang dapat dilihat pada tabel di atas.

Pada fitur *i-pronoun*, *i-propernoun*, *j-pronoun*, dan *j-propernoun*, pasangan *i* dan *j* akan di lakukan pengecekan berdasarkan tag yang telah diberikan oleh proses *POS Tagging*. Misalkan kata ***Ia*$_1$** (*j*) memiliki tag **"PRP",** maka *i-pronoun* diberi nilai 1 dan *i-propernoun* diberi nilai 0. Untuk kata ***Zaenab*$_1$** (*i*) memiliki tag **"NNP",** maka *j-pronoun* diberi nilai 0 dan *j-propernoun* diberi nilai 1.

Fitur ***Distance Of Words*** **(DOW)** digunakan untuk mencari nilai jarak antar kata, jarak kata-*i* dan kata-*j* dimulai dari awal kata pada teks. Sebagai contoh yaitu kata-*i* **zaenab** dengan lokasi kata = 12 dan kata-*j* **ia** dengan lokasi kata = 5. Adapun rumus yang digunakan sebagai contoh berikut.

$$DOW = \frac{\text{Lokasi Kata } i - \text{Lokasi Kata } j}{\text{Jumlah kata dalam dokumen teks}} = \frac{12-5}{38} = 0.18$$

Fitur ***Distance Of Sentence*** **(DOS)** digunakan untuk mencari nilai jarak antar kalimat, jarak kalimat *i* dan *j* dimulai dari awal kalimat pada teks,. Sebagai contoh yaitu kata-*i* **zaenab** dengan lokasi kalimat = 1 dan kata-*j* **ia** dengan lokasi kalimat = 1. Adapun rumus yang digunakan sebagai berikut.

$$DOS = \frac{\text{Lokasi Kalimat } i - \text{Lokasi Kalimat } j}{\text{Jumlah kalimat dalam dokumen teks}} = \frac{1-1}{4} = 0$$

Berikut contoh tahap ekstraksi fitur dari data training yang sudah melalui tahap *Preprocessing* .

Tabel 7 Contoh Ekstraksi Fitur Data *Training*

No	I		J		Fitur i		Fitur j		Relasi		
	kata-i	tag-i	kata-j	tag-j	i *pron*	i *prop*	j *pron*	j *prop*	DOW	DOS	*String Match*
1	zaenab$_1$	NNP	ia$_1$	PRP	0	1	1	0	0.18	0	0
2	kita$_2$	PRP	ia$_1$	PRP	1	0	1	0	0.26	0.25	0
3	kita$_2$	PRP	zaenab$_1$	NNP	1	0	0	1	0.08	0.25	0
4	kita$_3$	PRP	ia$_1$	PRP	1	0	1	0	0.37	0.5	0
5	kita$_3$	PRP	zaenab$_1$	NNP	1	0	0	1	0.18	0.5	0
6	kita$_3$	PRP	kita$_2$	PRP	1	0	1	0	0.11	0.25	1
7	zaenab$_3$	NNP	ia$_1$	PRP	0	1	1	0	0.58	0.5	0
8	zaenab$_3$	NNP	zaenab$_1$	NNP	0	1	0	1	0.39	0.5	1
9	zaenab$_3$	NNP	kita$_2$	PRP	0	1	1	0	0.32	0.25	0
10	zaenab$_3$	NNP	kita$_3$	PRP	0	1	1	0	0.21	0	0
11	kufah$_4$	NNP	ia$_1$	PRP	0	1	1	0	0.76	0.75	0
12	kufah$_4$	NNP	zaenab$_1$	NNP	0	1	0	1	0.58	0.75	0
13	kufah$_4$	NNP	kita$_2$	PRP	0	1	1	0	0.5	0.5	0
14	kufah$_4$	NNP	kita$_3$	PRP	0	1	1	0	0.39	0.25	0
15	kufah$_4$	NNP	zaenab$_3$	NNP	0	1	0	1	0.18	0.25	0
16	zaenab$_4$	NNP	ia$_1$	PRP	0	1	1	0	0.87	0.75	0
17	zaenab$_4$	NNP	zaenab$_1$	NNP	0	1	0	1	0.68	0.75	1
18	zaenab$_4$	NNP	kita$_2$	PRP	0	1	1	0	0.61	0.5	0
19	zaenab$_4$	NNP	kita$_3$	PRP	0	1	1	0	0.5	0.25	0
20	zaenab$_4$	NNP	zaenab$_3$	NNP	0	1	0	1	0.29	0.25	1
21	zaenab$_4$	NNP	Kufah$_4$	NNP	0	1	0	1	0.11	0	0

2.9 Klasifikasi SVM

Pada tahap klasifikasi, setiap pasangan kata *i* dan *j* akan diberi nilai positif 1 atau negatif 1 untuk dijadikan label (y_i). Nilai dari label tersebut digunakan untuk menyatakan *coreference* jika positif, dan tidak *coreference* jika negatif. Pada tahap pengklasifikasian ini menggunakan metode SVM, tahap yang akan dilakukan adalah untuk data latih terlebih dahulu kemudian akan dilakukan atau pengujian.

2.9.1 *Training* SVM

Pada tahap ini, hasil dari ektraksi fitur diubah menjadi format SVM, dimana data yang digunakan sebagai data latih yaitu novel.txt dan juga lebel.txt, dimana label digunakan untuk memberikan pembelajara untuk data latih, atau untuk mengetahui relasi kata yang memiliki kesetaraan yang sama, sehingga pada data latih akan menghasilkan kelas positif apabila memiliki label *coreference* yang sama, dan akan menjadi kelas negative apabila label *coreference* berbeda. Dan hasil dari *training* akan disimpan sebagai model svm untuk digunakan sebagai prediksi untuk data testing.

2.9.2 *Testing* SVM

Pada tahap Testing, akan dilakukan suatu prediksi dimana model yang dihasilkan dari data training akan digunakan. Sehingga data testing dakan mengklasifikasi dengan menggunakan model dari data training. pada penelitian ini, klasifikasi yang dilakukan menggunakan LIBSVM, yaitu library SVM untuk PHP dari Gabriel Coutco yang dapat dikunjugi pada GITHUB-nya.

2.10 Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan sistem berjalan sesuai dengan rancangan sistem yang telah dibuat sebelumnya. Pengujian sistem meliputi pengujian fungsionalitas dan pengujian akurasi.

2.10.1 Pengujian Fungsionalitas

Pengujian *black box* digunakan untuk menguji fungsi-fungsi khusus dari perangkat lunak yang dirancang. Berikut adalah tabel rencana pengujian black box.

Tabel 8 Rencana Pengujian *Black Box*

No	Aktifitas	Keluaran yang diharapkan	Jenis pengujian
1	Case Folding	Teks novel dengan huruf kecil	*Black Box*
2	Filtering	Teks novel tanpa karakter selain “a-z”,”.”,”-”, dan spasi	*Black Box*
3	Tokenisasi Kalimat	Kumpulan kalimat tanpa delimeter titik	*Black Box*
4	Tokenisasi Kata	Kumpulan kata	*Black Box*
5	POS Tagging	Kumpulan Kata dasar beserta Tag setiap kata yang diberikan	*Black Box*
6	Penentuan Pasangan Kata *Training*	Pasangan kata *i* dan *j*	*Black Box*
7	Penentuan Pasangan Kata Testing	Pasangan kata *i* dan *j*	*Black Box*
8	Penentuan Nilai Ekstraksi Fitur	Kumpulan pasangan *coreference* dan hasil ekstraksi fitur berupa angka	*Black Box*
9	Training SVM	Model SVM	*Black Box*
10	Testing SVM	Hasil dari Klasifikasi SVM berupa pasangan *coreference* yang memiliki kelas.	*Black Box*

2.10.2 Pengujian Akurasi

Pengujian akurasi pada penelitian ini dilakukan untuk mengetahui hasil akurasi yang didapatkan dari sistem *Coreference Resolution* dengan menambahkan pendeteksian untuk kata ganti jamak dan dengan menggunakan metode SVM pada teks berbahasa Indonesia.

Data yang digunakan sebagai data latih yaitu dari 8 novel berbahasa Indonesia yang terdapat 126 kata ganti yang terdiri dari nama orang, kata ganti orang(tunggal dan jamak), jumlah pasangan kata yang terbentuk adalah 7875 pasangan. Teks atau cerita yang dijadikan sebagai data latih memiliki pola yang beragam, hal ini akan mempengaruhi hasil klasifikasi karena hasil dari ekstraksi fitur akan memiliki banyak pola.

Berdasarkan hasil deteksi pasangan kata yang telah dianalisis dan dibandingkan dengan hasil sistem, pasangan kata terbentuk dengan sempurna . Pengujian klasifikasi dari sistem *Coreference Resolution* dengan menggunakan 10 data uji yang diambil dari 5 novel dan dengan menggunakan metode SVM, maka menghasilkan akurasi sebagai berikut.

Tabel 9. Hasil Pengujian

Judul Novel	File	Jumlah Pasangan Kata	Jumlah	
			Benar	Salah
5 Cm	cm1.txt	55	30	25
	cm2.txt	36	24	12
Dear Nathan	dear1.txt	45	34	11
	dear2.txt	78	46	32
Garis Waktu	garis1.txt	45	25	20
	garis2.txt	45	23	22
Rindu	rindu1.txt	55	33	22
	rindu2.txt	66	43	23
Teman Hati	teman1.txt	36	27	9
	teman2.txt	91	54	37
Jumlah		552	339	213

Berikut adalah perhitungan untuk nilai akurasi keseluruhan pada klasifikasi novel dengan metode SVM.

$$Akurasi = \frac{Jumlah\ yang\ diklasifikasi\ secara\ benar}{Jumlah\ sample\ pasangan\ kata\ yang\ diuji}\ x\ 100\%$$

$$= \frac{339}{552}\ x\ 100\%$$
$$= 61.41\ \%$$

Berdasarkan tabel diatas, hasil pengujian dari sistem *Coreference Resolution* menggunakan metode SVM menghasilkan Akurasi 61.41%

1. Pola data latih yang dihasilkan masih terdapat kekurangan untuk beberapa pola cerita dan beberapa kata ganti sehingga mengklasifikasi tidak sesuai seperti berikut.
   a. Pengenalan kata ganti **kau** pada file garis2.txt
   b. Pemaparan cerita pada file cm2.txt, terdapat pola yang menyebutkan kata jamak secara berdekatan seperti kata **kalian** dan **kita** yang memiliki objek *coreference* yang berbeda.
   c. Kata ganti yang tampil terlebih dahulu sebelum entitas bernama banyak mengalami kesalahan dalam klasifikasi.
2. Belum dilakukannya pendeteksian untuk kata ganti kepunyaan seperti kata **budakmu**, karena kata kepunyaan sering muncul dan apabila dipisahkan, kata **budak** akan menjadi terdeteksi sebagai kata ganti, dan kata **mu** dapat terdeteksi sebagai PRP yang dapat dijadikan kandidat *coreference*.

3 PENUTUP

Pada bagian ini akan menjelaskan kesimpulan dan saran dari penelitian.

3.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat diketahui bahwa *Coreference Resolution* bahasa Indonesia yang mengandung entitas jamak dapat dilakukan dan digunakan sebagai pendeteksi kata ganti untuk orang yang dijadikan sebagai *coreference*. Pada penelitian ini menggunakan 8 data latih berupa kutipan novel yang membentuk 7875 pasangan kata dan 10 data uji dengan total 552 pasangan kata yang mengandung entitas jamak, kata ganti, dan entitas bernama. Akurasi yang dihasilkan dari penelitian ini sebesar 61.41%.

3.2 Saran

Berdasarkan hasil pengujian yang telah dilakukan, maka saran yang diharapkan pada pengembangan selanjutnya untuk sistem *Coreference Resolution* yaitu.

1. Memaksimalkan kinerja dari POS *Tagging* untuk dapat mengenali jenis kata ganti yang memiliki kata imbuhan.
2. Memaksimalkan kinerja SVM dalam menemukan *hyperlane* terbaik untuk penanganan *Coreference Resolution.*
3. Menambahkan proses *Rule Based* untuk dapat mendeteksi kata ganti kepemilikan yang dapat dijadikan sebagai *coreference.*
4. Menambahkan fitur *Named Entity Recognation* (NER), untuk dapat mengenali entitas bernama seperti nama orang, nama organisasi dan nama tempat.

DAFTAR PUSTAKA

[1] J. Gao, F. Kong, P. Li, and Q. Zhu, “Research of noun phrase coreference resolution,” *Proceedings. 2011 International Conference Asian Language. Processing.* 2011.

[2] Kamus Besar Bahasa Indonesia. 2017. https://www.kbbi.web.id/, 20 Februari 2018 15.30

[3] Ayu Linggar Sari, “Coreference Resolution Menggunakan Metode SVM Pada Novel Berbahasa Indonesia,” Perpustakaan Unikom 2017.

[4] L. Assaffat, “Analisis Akurasi Support Vector Machine Dengan Fungsi Kernel Gaussian Rbf Untuk Prakiraan Beban Listrik Harian Sektor Industri,” *Momentum*, 2015.

[5] A. D. Putri, “Klasifikasi Dokumen Teks Menggunakan Metode Support Vector Machine Dengan Pemilihan Fitur Chi-Square,” *skripsi. Inst. Pertan. Bogor, Bogor*, 2013.