SRITI2013

by Viny M

Submission date: 08-Nov-2020 03:27PM (UTC+0700)
Submission ID: 1439416536
File name: New_Paper_Viny.pdf (290.1K)
Word count: 3988
Character count: 24194

Implementasi Stanford NER untuk Pemberian Entitas pada Dokumen Bahasa Indonesia

Viny Christanti M. , M.Kom [1)] , Ir. Jeanny Pragantha, M.Eng [2)] dan Andreas Aditya [3)]
[1,2,3)] *Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara*
Jl. Let. Jend. S. Parman no. 1, Jakarta, 11440
021-5671747
[1)] **E-mail :** viny@untar.ac.id

Abstrak

Named Entity Recognition (NER) adalah suatu teknik yang dapat digunakan untuk memberikan suatu label kata (entitas) tertentu pada suatu data teks. Entitas yang dimaksud dapat berupa nama orang, lokasi, organisai, nama hari dan lainnya. Beberapa metode yang digunakan dalam *NER* adalah metode statistik yang terdiri dari *hidden markov model*, *maximum entropy*, dan *conditional random field*. Metode tersebut sudah diterapkan pada sistem NER terkemuka yang telah dibuat adalah *Stanford NER, Lingpipe, GATE* dan lain-lain. Penelitian ini mengimplementasikan metode *Conditional Random Field* (CRF) yang sudah dikembangkan oleh Stanford untuk bahasa Inggris. Tujuan dari implementasi ini adalah untuk merancang sebuah program yang dapat membantu proses pemberian entitas terhadap dokumen bahasa Indonesia berdasarkan Stanford NER.

Pengujian digunakan dengan 10 dokumen berita yang digunakan pada proses testing, yang terdiri dari 5.279 kata. Hasil pemberian entitas dengan program aplikasi menunjukkan tingkat keakurasian sebesar 59%. Fitur yang memberikan nilai akurasi tertinggi adalah fitur Current Word yaitu fitur yang melihat hanya pada kata yang diobservasi sesuai dengan data training yang disediakan.

Kata Kunci : Conditional Random Field, Entitas Bahasa Indonesia, Named Entity Recognition, Natural Language Processing, Stanford NER

1. Pendahuluan

Seiring dengan perkembangan teknologi, setiap orang dituntut supaya dapat memanfaatkan perkembangan itu dalam kehidupan sehari-hari. Perkembangan teknologi mencakup semua aspek kehidupan, salah satunya dalam bidang bahasa. Bahasa memiliki peranan yang sangat penting dalam pertukaran informasi dan atau menerima informasi. Salah satu media untuk pertukaran informasi adalah melalui membaca. Untuk mendapatkan informasi yang tepat dari membaca, diperlukan pengetahuan tentang sebuah tata bahasa yang baik dan benar.

Salah satu pemanfaatan teknologi dalam bidang bahasa adalah adanya program *NER (Named Entity Recognition)*. *NER* adalah kegiatan pemberian label kata pada suatu kata [1]. NER dapat dianggap sebagai proses klasifikasi kata ke dalam kategori yang sesuai. Pada umumnya NER fokus dalam mengklasifikasi kategori seperti nama orang, lokasi serta organisasi. Hasil dari pemberian NER ini diaplikasikan untuk sistem yang lebih besar misalnya *Question and Answering Systems*, *Search Engine* atau *Machine Translator*.

Beberapa sistem NER terkemuka yang telah dibuat adalah *Stanford NER* [1], *Lingpipe*[2], *GATE*[3] dan lain-lain. Pada umumnya sistem NER yang ada dibuat hanya untuk pengembangan atau digabungkan dengan sistem lainnya. Sehingga tidak tersedia dalam bentuk yang mudah untuk digunakan. Salah satunya adalah Stanford NER, yang tersedia dalam 2 versi. Versi pertama menggunakan GUI dan yang kedua adalah dalam bentuk *class* yang harus digabungkan dalam program lain atau dalam bentuk *console*. Bentuk dalam GUI, hanya diperuntukkan untuk mencoba hasil training yang sudah ada. Kelemahan bentuk GUI ini adalah tidak dapat diimplementasikan untuk bahasa lain.

Penelitian di bidang *NER* untuk bahasa Indonesia sudah banyak dilakukan namun belum menghasilkan hasil yang maksimal. Hal ini karena bahasa Indonesia memiliki

[1] http://nlp.stanford.edu/software/CRF-NER.shtml
[2] http://alias-i.com/lingpipe/
[3] www.gate.ac.uk

aturan yang berbeda dan memiliki ambiguitas lebih besar dibandingkan dengan bahasa lain, misalnya bahasa Inggris, Aplikasi *NER* yang sudah ada, sebagian besar memang dibangun dengan bahasa Inggris sehingga tidak dapat langsung diaplikasikan untuk bahasa Indonesia [2].

Stanford NER merupakan sebuah sistem dengan metode *Conditional Random Field* (CRF) yang digunakan untuk memberikan entitas pada kata secara otomatis. Seperti diketahui, Stanford NER terdiri dari dua tahapan, yaitu tahapan training dan testing. Training dilakukan untuk membentuk classifier yang sesuai dengan karakteristik bahasa masing-masing. Stanford NER sendiri telah menyediakan beberapa classifier untuk bahasa Arab, Cina dan Jerman. Sedangkan bahasa Indonesia belum disediakan oleh Stanford NER. Classifier yang sudah tersedia dapat digunakan dengan mudah untuk melakukan pemberian label terhadap setiap kata. Classifier tersebut dapat langsung digunakan pada GUI Stanford NER yang sudah tersedia atau digabungkan dalam program lain.

Dalam melakukan training dibutuhkan beberapa tahapan yaitu menyiapkan dokumen, melakukan tokenisasi, pemberian label secara manual dan pemilihan fitur. Tahapan ini membutuhkan waktu yang tidak sedikit. Stanford NER pun menyediakan berbagai macam fitur yang dapat disesuaikan dengan karakteristik bahasa. Fitur tersebut antara lain melihat kata sebelumnya, melihat susunan huruf besar dan kecil, melihat kata sesudahnya dan kata-kata yang khusus.

Proses yang tidak mudah dalam membangun classifier membuat setiap penelitian harus melakukan proses training dari awal. Pada tulisan ini masalah tersebut diatasi dengan memimplementasikan Stanford NER dalam bentuk sebuah program dengan user interface sehingga mempermudah proses training dan testing Stanford NER. Banyaknya fitur yang ada dalam Stanford NER, dapat menyulitkan peneliti untuk menemukan fitur yang tepat. Dengan adanya program ini, diharapkan dapat membantu peneliti untuk memilih fitur tersebut.

Permasalahan yang muncul dalam perancangan ini adalah bagaimana sistem dapat membaca isi dokumen, bagaimana sistem akan melakukan proses *training* dokumen, bagaimana sistem akan mengimplementasikan fitur-fitur yang dipilih, dan bagaimana sistem melakukan pemberian *entitas* dengan Conditional Random Field (*CRF)*. Tujuan perancangan ini adalah merancang sebuah sistem untuk mempermudah proses pemberian entitas bahasa Indonesia dan menemukan fitur yang tepat dari suatu kata. Sehingga pada penelitian ini, classifier yang terbentuk difokuskan pada dokumen bahasa Indonesia.

2. Natural Language Processing

Natural Language Processing (NLP) atau pengolahan bahasa alami merupakan salah satu bidang ilmu *Artificial Intelligence* (kecerdasan buatan) yang mempelajari komunikasi antara manusia dengan komputer melalui bahasa alami. Pemrosesan bahasa alami tidak mudah dilakukan. Beberapa alasan yang menyulitkan pemrosesan bahasa alami diantaranya adalah dalam bahasa alami sering terjadi ambiguitas atau makna ganda, jumlah kosa kata (*vocabulary*) dalam bahasa alami sangat besar dan berkembang dari waktu ke waktu [3]. Beberapa tingkatan dari *natural language processing* adalah [5]:

1. Fonologi yang berhubungan dengan interprestasi bunyi ujaran dalam dan di antara kata-kata.
2. Morfologi yang berkaitan dengan sifat komponen makna dari kata-kata, yang terdiri dari morfem.
3. Leksikal yang melibatkan identifikasi pengolahan kata dan menentukan kelas tata bahasa yang nantinya digunakan pada tingkat sintaksis.
4. Sintaksis yang berfokus pada menganalisis kata-kata dalam sebuah kalimat untuk mengungkapkan struktur gramatikal pada kalimat.
5. Semantik berkaitan dengan pengertian yang bebas dengan konteks, mengambil satu kalimat pada suatu waktu.
6. Pragmatik yang berhubungan dengan pengetahuan yang berkaitan dengan masing-masing konteks yang berbeda tergantung pada situasi dan tujuan pembuatan sistem.

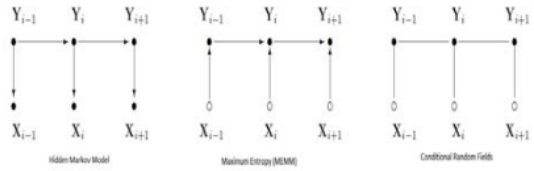
Pemberian entitas sendiri berada pada posisi leksikal. Dimana dengan adanya pemberian label pada setiap kata dapat membantu member informasi kepada komputer mengenai makna kata tersebut. Hasil pemberian entitas ini akan diteruskan untuk memahami struktur kalimat pada level selanjutnya.

2.1. Named Entity Recognition

Named Entity Recognition (NER) adalah proses memberi label atau *entitas* pada setiap kata dalam kalimat dengan *entitas* yang sesuai untuk kata tersebut [6]. Pemberian *entitas* dapat dimanfaatkan pada aplikasi NLP lain, seperti *information extraction*. Penggunaan *NER* dapat membantu mencari suatu kata yang penting dalam suatu dokumen. *NER* dapat dilakukan secara manual maupun otomatis. *NER* dilakukan secara manual dengan menggunakan bantuan satu atau beberapa ahli bahasa yang memberikan *entitas* yang bersesuaian untuk tiap kata pada suatu teks atau *corpus* [4]. Beberapa metode yang digunakan dalam *NER* adalah metode statistik yang terdiri dari *hidden markov model*, *maximum entropy*, dan *conditional random field*.

Perbedaan ketiga metode ini hanya akan mempengaruhi keakuratan dan kemampuan sistem program dalam memberikan entitas yang sesuai. Untuk lebih jelasnya dapat dilihat pada **gambar 1** di bawah ini,

Berdasarkan gambar 1, dapat dilihat bahwa metode *Conditional Random Field* (*CRF*) merupakan metode yang bersifat tidak berarah, sehingga metode ini dapat dengan cepat dalam memberikan entitas namun dengan akurasi yang tidak buruk. Selain dipengaruhi oleh metode yang digunakan, tingkat keakuratan sebuah pemberian *entitas* juga dipengaruhi oleh beberapa faktor yaitu: jumlah data yang digunakan saat *training* dan perbedaan antara *corpus* (teks) yang digunakan pada saat *training* dengan saat menggunakan aplikasi, serta jumlah *unknown words* (kata yang tidak dikenali) [4].



Gambar 1 Perbedaan Graph HMM, MEMM dan CRF

Stanford NER merupakan sebuah sistem dengan metode *Conditional Random Field* (CRF) yang digunakan untuk memberikan entitas pada token seperti nama orang, nama organisasi, atau nama tempat. Stanford NER dikembangkan oleh The Stanfords Natural Language Processing Group dari Universitas Stanford. Stanford NER mulai dikembangkan oleh NLP group pada tahun 2003 [7].

2.2. Conditional Random Field

Conditional Random Field (CRF) merupakan metode pemberian *entitas* yang diperkenalkan oleh John Lafferty pada tahun 2001 [7]. Metode ini didasarkan pada ilmu statistika yang mengutamakan probabilitas bersyarat. Metode ini merupakan metode dengan model graph tidak berarah yang memungkinkan adanya evaluasi antar kata dengan kemungkinan yang sangat banyak. *CRF* sebagai salah satu model kondisional juga memilik makna bahwa *CRF* bekerja dengan konsep probabilitas kondisional terhadap suatu rangkaian label, berdasarkan sebuah rangkaian observasi. Sehingga nilai probabilitas suatu rangkaian label yang menjadi keluaran *CRF* bergantung pada rangkaian observasi yang menjadi input *CRF*.

CRF merupakan sebuah kerangka untuk membangun sebuah model probabilistik yang digunakan untuk melakukan proses segmentasi dan pelabelan data. *CRF* sendiri memiliki bentuk berupa model graf tidak berarah yang berarti setiap sisi yang menghubungkan setiap titik dalam suatu graf tidak memiliki arah. Pada *CRF*, setiap titik pada graf merepresentasikan sebuah variabel acak dan setiap sisinya merepresentasikan hubungan antar dua variabel acak.

CRF menyatakan sebaran log-linier untuk sebuah rangkaian label berdasarkan sebuah rangkaian observasi. Rangkaian observasi merupakan rangkaian simbol yang "dilihat" oleh *CRF* dan menjadi masukan bagi proses probabilistik yang dilakukan *CRF*. Sementara rangkaian label merupakan keluaran dari *CRF* berupa satu untaian simbol yang dihasilkan dari proses probabilistik *CRF*.

Berikut ini adalah contoh perumusan dari metode CRF itu sendiri,[7]

$$P_\theta(Y|X) \propto \exp\left(\sum_{i=1,k}\lambda_k f_k(e_i, Y|_{e_i}, X) + \sum_{i=1,k}\lambda_k g_k(u_i, Y|_{u_i}, X)\right) \quad ...(2)$$

Pada keterangan di atas f_k dan g_k merupakan fungsi fitur. f_k adalah fitur sisi yang berurusan dengan transisi antara label dalam suatu rangkaian label. Sementara g_k merupakan fitur titik yang berurusan dengan label individu dalam suatu rangkaian. λ_k dan μ_k merupakan parameter yang nilainya diperkirakan berdasarkan data yang digunakan dalam proses pelatihan. Perkiraan parameter digunakan untuk mendapatkan nilai $\theta = (\lambda_1, \lambda_2, ...; \mu_1, \mu_2, ...)$ yang dapat memaksimalisasi nilai probabilitas rangkaian observasi.

2.3. Stanford NER

Stanford NER merupakan salah satu sistem pemberian NER yang sudah banyak dipakai oleh para peneliti. Untuk mengimplementasikan Stanford NER terhadap bahasa lain, ada 2 tahap yang harus dikerjakan yaitu tahapan training dan testing. Tahap training dilakukan untuk membangun classifier. Setelah classifier terbentuk tahapan selanjutnya adalah testing.

Classifier merupakan sebuah alat pembelajaran mesin yang mengambil informasi dan menentukannya ke dalam salah satu dari *k*-buah kelas [8]. Pelatihan dilakukan dengan beberapa langkah antara lain ekstraksi fitur dan perhitungan bobot untuk tiap fitur. Fitur-fitur yang dapat digunakan dalam melatih sebuah *classifier* adalah [9] :

1. Current Word: Fitur ini mencari keseluruhan input data dan akan mengenali apa saja kata yang terdapat pada data training yang sesuai dengan entitasnya.
2. Previous Word: Fitur ini akan mencari suatu kata yang telah sesuai dengan entitasnya, kemudian akan membandingan kata tersebut dengan kata sebelumnya yang telah di-training sebelumnya. Fitur ini juga secara signifikan akan meningkatkan tingkat akurasinya. Contoh: biasanya setelah tanda titik, kata selanjutnya adalah kata orang, yang termasuk ke dalam entitas person.
3. Next Word: Fitur ini berfungsi untuk mengecek apakah kata yang telah diberi label telah sesuai dengan kata selanjutnya. Contoh: biasanya

setelah nama orang maka kata selanjutnya biasanya adalah kata kerja.

4. Current Word Character n-gram, (WordNgram): Fitur ini berfungsi untuk menghitung nilai probabilistik ngram suatu kata. Ngram yang digunakan adalah N=2.
5. Current POS Tag: Fitur yang berfungsi untuk membandingkan kata yang telah diberi label dengan bantuan POS Tagging (Part Of Speech Tagging).
6. Surrounding POS Tag Sequence, (Sequence): Fitur ini berfungsi untuk membandingkan sebuah kata yang telah diberi tagging dengan kata-kata yang berada disekitarnya.
7. Current Word Shape, (Wordshape): Fitur ini berfungsi untuk mengecek bentuk dari suatu kata, apakah kata tersebut merupakan suatu bentuk lain dari kata yang sama.
8. Surrounding Word Shape Sequence: Fitur ini mempunyai fitur yang sama dengan current word shape, hanya bedanya, fitur ini mengecek kata-kata yang berada si antara kata yang akan diberi label.
9. Fitur Class: Fitur ini berguna untuk mengelompokan setiap kata dengan entitasnya masing-masing.
10. No Mid NGram: Fitur ini berguna untuk menghilangkan nilai tengah dari Ngram suatu kata.

Algoritma CRF pada Stanford NER itu sendiri adalah proses inisialisasi yang terdiri dari pemberian *entitas* secara manual dan pemberian *entitas* secara automatis, serta fase pembelajaran yang terdiri dari pengulangan dalam menghitung nilai kesalahan dari setiap kata dengan *entitas*, memilih fitur yang terbaik, dan mengulangi (evaluasi kata) sampai mencapai nilai yang paling besar. Penggunaan algoritma CRF dalam Stanford NER, diharapkan juga dapat diimplementasikan pada baha Indonesia.

2.4. Rancangan dan Pembuatan Program

Program aplikasi yang dirancang bertujuan untuk memudahkan penggunaan Stanford NER untuk bahasa Indonesia dan memberikan *entitas* secara otomatis pada suatu kata. Program aplikasi dibagi menjadi 2 tahap yaitu training dan testing. Tahapan dalam proses training adalah sebagai berikut:

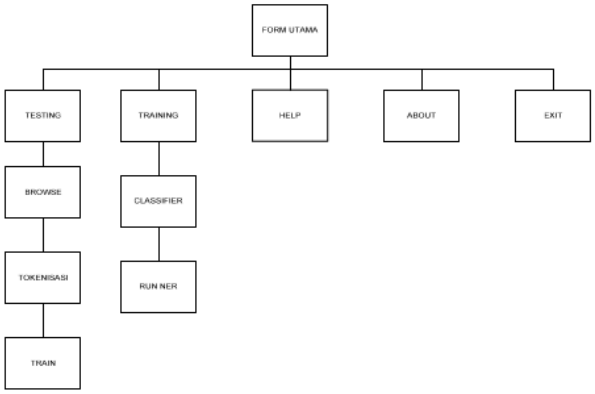
a. Menyiapkan *corpus* (teks) yang belum diberi entitas.
b. Tokenisasi *corpus* yang belum diberi entitas menjadi dua kolom, pada kolom pertama berisi kata dari *corpus*, kolom kedua berisi entitas dari masing-masing kata.
c. Memberikan entitas pada setiap kata secara manual, kata yang tidak mempunyai entitas diberikan huruf "O" (*Other*).
d. Membuat daftar fitur yang akan digunakan.
e. Melatih *Classifier* dengan menggunakan *corpus* yang sudah diberi entitas.

Sedangkan tahapan proses testing adalah sebagai berikut:

a. Menyiapkan file yang akan dicari entitas nya.
b. Memilih *classifier* yang akan digunakan
c. Melakukan proses testing dengan program.

Data yang diinput pada aplikasi ini berupa dokumen artikel berita berbahasa Indonesia yang telah diubah ke dalam bentuk teks dalam format .txt. Jumlah dokumen yang digunakan pada perancangan ini adalah sebanyak 60 dokumen artikel berita. Terdiri dari 50 dokumen artikel untuk proses *training* dan 10 dokumen artikel untuk proses pengujian. Sementara itu, untuk jumlah kata yang diproses pada sebuah dokumen, tidak dibatasi.

Perancangan diagram hirarki bertujuan untuk mendapatkan gambaran mengenai modul yang dibuat. Rancangan diagram hirarki dapat dilihat pada **Gambar 2**. Tampilan pertama dalam program aplikasi ini adalah menu utama yang menampilkan lima tombol menu yang mengarah ke modul-modul yang dapat dipilih pengguna, yaitu: modul *training*, modul *testing*, modul *about*, modul *help* dan modul *exit*.



Gambar 2 Rancangan Diagram Hirarki

Pembuatan sistem diawali dengan membuat rancangan sistem yang digunakan. Setelah itu dilakukan tahap pembuatan program aplikasi yang dimulai dari pembuatan GUI (*graphical user interface*) sampai dengan pengujian hasil dan evaluasi hasil pemberian *entitas* dari

program yang dirancang. Spesifikasi dari perangkat keras yang akan digunakan dalam pembuatan program aplikasi ini antara lain:

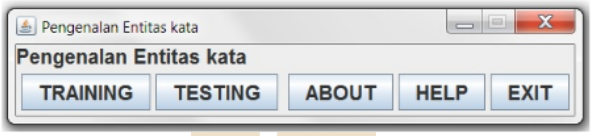
1. cessor Intel(R) Core(TM)2 Duo T5550 1.83 Ghz
2. Hard disk berkapasitas 160 GB
3. Memori RAM 512 MB
4. Monitor 12.1”
5. Keyboard
6. Optical mouse
7. DVD ROM

Spesifikasi dari perangkat lunak yang akan digunakan dalam pembuatan aplikasi ini antara lain:

1. Microsoft Windows 7 Home Premium
2. Netbeans 7.1
3. Java Development Kit 1.6
4. Wordpad

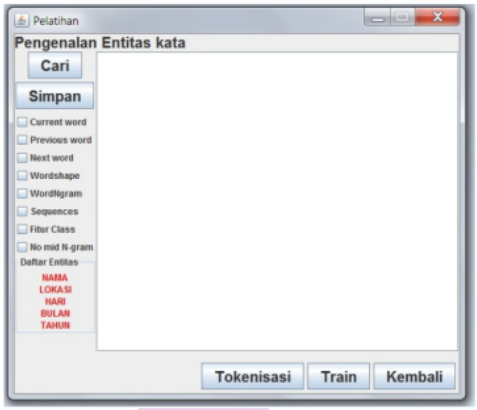
Program aplikasi dibuat dengan menggunakan bahasa pemrograman Java dengan Netbeans. Tahap-tahap dalam membuat program adalah :

1. *Form* Utama
   Form utama merupakan *form* awal pada program yang berhubungan dengan *form-form* lainnya. Didalam form utama terdapat *button training*, *button testing*, *button about*, dan *button help,* dan *button exit*. Form utama dapat dilihat pada **gambar 3**.



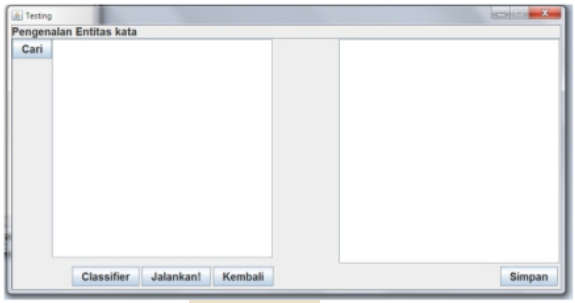
Gambar 3 *Form* Utama

2. *Form Training*
   Pada *form* ini terdapat 5 buah tombol utama dan 8 buah fitur yang dapat dipilih. *Button Cari* untuk memilih file yang akan digunakan untuk proses *training*. *Button Simpan* digunakan untuk menyimpan file yang sudah di-*load* oleh program. *Button Tokenisasi* digunakan untuk memecah setiap kata yang ada didalam sebuah file menjadi 1 buah kata per-baris dan menambahkan "O" di akhir kata tersebut. *Button Train* digunakan untuk membu*classifier*. *Button Kembali* digunakan untuk kembali ke *form* utama. Form *training* dapat dilihat pada **gambar 4**.



Gambar 4 *Form Training*

3. *Form Testing*
   Form ini berisi *5* buah tombol yang dapat digunakan untuk mengoperasikan fungsi utama. *Button Cari* berguna untuk membuka file yang akan dicari *entitas*nya. *Button Classifier* digunakan untuk me-*load classifier* yang dibuat pada saat menggunakan *form* training ataupun *file classifier* yang telah disertakan. *Button Jalankan* berguna untuk memulai proses pencarian *entitas* secara automatis (program). *Button kembali* digunakan untuk kembali ke *form* utama. *Button Simpan* digunakan untuk menyimpan hasil nberian *entitas* secara automatis. Form *testing* dapat dilihat pada **gambar 5**.



Gambar 5 *Form Testing*

3. Hasil Pengujian

Tahap-tahap dalam pengujian sistem, antara lain :

1. Mengumpulkan dokumen artikel yang digunakan sebagai bahan pengujian program. Dokumen artikel didapat dari *website* berita seperti www.kompas.com. Artikel berisi berita tentang seputar olahraga dan kesehatan. Artikel yang didapat dari hasil pencarian dokumen sebanyak 60 buah dokumen artikel. Sebanyak 50 dokumen

digunakan untuk proses *training* dan 10 dokumen digunakan untuk proses *testing*.

2. Melakukan pengujian terhadap setiap modul dan tombol untuk mengecek apakah semua modul dan tombol yang terdapat pada program berjalan dengan baik sesuai dengan fungsinya masing-masing.
3. Melakukan proses *training* dan *testing*.
4. Membuat buku *manual* yang bertujuan membantu pengguna menggunakan program aplikasi ini.

Pengujian keseluruhan terhadap aplikasi ini dilakukan dengan menjalankan *form-form* yang tersedia, yaitu *form* utama, *form training*, form *testing*, form *about* dan *form help*. Pengujian terhadap seluruh *form* dapat dikatakan berhasil karena seluruh *form* berjalan sebagaimana mestinya. Semua menu dan tombol dalam masing-masing *form* dapat menjalankan fungsinya dengan baik**.** Contoh hasil pemberian *NER* pada program dapat dilihat pada **gambar 6** dan contoh artikel dapat dilihat pada **gambar 7.**

Gambar 6 Hasil Pemberian Entitas

```
Sementara <nama>Wanggai</nama> terlihat sibuk "melobi"
<lokasi>Keane</lokasi> di depan bangku cadangan. Dari pengamatan
Kompas.com di tribun media, <nama>Oktovianus Maniani</nama> juga
terlihat menginginkan kaus mantan pemain <nama>Tottenham
Hotspur</nama> itu.

Namun, <nama>Keane</nama> terlihat tak kunjung membuka kausnya
sampai ia akhirnya menuju ruang ganti. Ternyata, usaha
<nama>Wanggai</nama> tak sia-sia. Penyerang asal
<lokasi>Papua</lokasi> itu berhasil mendapatkan seragam nomor 14
milik <nama>Keane</nama>.
```

Gambar 7 Contoh Artikel yang sudah diberi label secara otomatis

Setelah dilakukan pengujian terhadap *form-form* yang ada, maka dilakukan pengujian terhadap pemberian *entitas*. Pada pengujian pertama, pelatihan menggunakan data training yang berisi 50 buah artikel berbahasa Indonesia dengan jumlah kata mencapai 32.685 buah kata. Sedangkan pada pengujian kedua, *file* data *training* berisi 50 buah artikel pada pengujian pertama, ditambah dengan daftar nama orang-orang Indonesia, lokasi yang ada di Indonesia, nama-nama hari, dan tahun. Pengujian dilakukan terhadap 10 dokumen artikel, dengan jumlah seluruh kata sebanyak 5.279 kata. Jumlah data dan kata dapat dilihat pada **tabel 1**.

Tabel 1. Jumlah data training dan testing pada setiap pengujian

Pengujian	Jumlah data training		Jumlah data testing	
	Dokumen	Kata	Dokumen	Kata
1	50	32.685	10	5.279
2	50+daftar nama, lokasi, hari, tahun di Indonesia	123.678	10	5.279

Penambahan daftar nama, lokasi, hari dan tahun yang berlaku di Indonesia ditujukan untuk mengatasi nama, lokasi, hari atau tahun yang tidak terdapat pada data training. Daftar tersebut diperoleh dari berbagai sumber seperti website, peta dan kamus bahasa Indonesia.

Pengujian dilakukan pada setiap fitur yang terdapat pada Stanford NER. Fitur tersebut dilakukan terhadap pengujian 1 dan 2. Hasil dari pengujian 1 dan 2 dapat dilihat pada **tabel 2**. Nilai akurasi didapat dari total entitas yang diperoleh oleh program, dibagi dengan jumlah entitas manual dari ke 10 buah file data *testing*.

Tabel 2 Hasil akurasi setiap fitur untuk pengujian 1 dan 2

Fitur	Pengujian 1	Pengujian 2
Current Words	50.51%	59%
Previous Words	54.76%	53%
Next Words	57.89%	54%
Word Shape	50.91%	55%
Word Ngrams	46.26%	58%
Sequence	50.91%	50%
Fitur Class	50.91%	52%
No midNgram	50.91%	54%
All fitur	45.34%	56%
Rata-rata	50.93%	54.35%

Pada **tabel 2**, dapat terlihat bahwa secara rata-rata pengujian 2 menghasilkan akurasi yang lebih baik dibandingkan akurasi pengujian 1. Sehingga terbukti pengaruh penambahan daftar nama mempengaruhi hasil akurasi walaupun tidak signifikan. Apabila dilihat dari fitur yang dipilih maka fitur Next Word merupakan fitur yang menghasilkan nilai akurasi terbaik yaitu 54.35%. Namun hal ini hanya berlaku pada pengujian 1. Hasil jumlah entitas yang benar diberi label dapat dilihat pada **tabel 3**. Pada tabel 3 juga dapat dilihat bahwa entitas organisasi paling banyak mengalami kesalahan diberi label.

Tabel 3 Jumlah entitas yang berhasil dan benar diberi label pada pengujian 1

Fitur	Entitas						Total	%
	Nama	Lokasi	Organisasi	Hari	Bulan	Tahun		
Entitas manual	316	132	247	12	5	35	988	
All fitur	312	91	0	11	0	34	448	45.34
No mid N-gram	438	30	0	0	0	35	503	50.91
Fitur class	438	30	0	0	0	34	503	50.91
Sequence	438	30	0	0	0	35	503	50.91
Word Ngram	305	106	0	11	0	35	457	46.26
Word Shape	438	30	0	0	0	35	503	50.91
Next Word	**360**	**170**	**0**	**8**	**0**	**34**	**572**	**57.89**
Previous Word	396	92	0	71	1	35	541	54.76
Current Word	366	91	0	7	0	35	499	50.51

Kesalahan pemberian label dapat terjadi karena pemilihan fitur yang kurang tepat dan kurang beragamnya data yang digunakan pada saat training. Pada **gambar 8** dapat dilihat bahwa kata <nama>Macau Terbuka Grand Prix Gold</nama> salah diberi label. Dimana seharusnya pemberian label untuk "Macau Terbuka Grand Prix Gold" adalah organisasi.

```
<lokasi>JAKARTA</lokasi>, KOMPAS.com  Dua tunggal putra
<lokasi>Indonesia</lokasi>, Taufik Hidayat dan <nama>Simon
Santoso</nama>, melangkah ke perempat final <nama>Macau Terbuka
Grand Prix Gold</nama>. Pada babak ketiga, <hari>Kamis</hari>
(1/12/2011), mereka bermain alot untuk menaklukkan lawannya
sehingga tetap memelihara peluang membawa pulang gelar turnamen
berhadiah 200.000 dollar AS tersebut.

Taufik, unggulan ketiga, dipaksa bermain rubber-game 21-11,
20-22, 21-16 melawan pemain <lokasi>China</lokasi>, <nama>Chen
Yuekun</nama>. Hal serupa juga dialami <nama>Simon</nama>,
unggulan kelima, yang bermain lebih dari satu jam untuk menang
21-19, 22-24, 21-12 atas unggulan ke-12 dari
<lokasi>Hongkong</lokasi>, <nama>Wong Wing Ki</nama>.

Di babak delapan besar, <hari>Jumat</hari> (2/12/2011), Taufik
bertemu unggulan keenam dari <lokasi>Korea</lokasi>, <nama>Lee
Hyun Il</nama>, yang menang 21-13, 21-15 atas unggulan ke-16 dari
<lokasi>Taiwan</lokasi>, <nama>Hsueh Hsuan Yi</nama>. Sementara
itu, <nama>Simon</nama> menghadapi unggulan ke-14 dari
<lokasi>India</lokasi>, <nama>Kashyap Parupalli</nama>, yang
menang 13-21, 21-17, 21-17 atas pemain <lokasi>Taiwan</lokasi>,
<nama>Chou Tien Chen</nama>.
```

Gambar 8 Contoh artikel hasil pemberian label secara otomatis untuk dokumen dengan judul "Taufik dan Simon Bertemu Pemain Taiwan"

Kesalahan pemberian label juga terjadi pada kalimat <organisasi>JAKARTA, Kompas.com - Taufik Hidayat lolos ke perempatfinal Macau Terbuka GP Gold</organisasi>. Kesalahan ini terjadi karena ada tanda "-" diantara kalimat tersebut yang dipisahkan oleh spasi. Padahal seharusnya kata "Jakarta" terpisah sendiri mendapat label <lokasi>. Kemudian kata "Taufik Hidayat" diberi label <nama> dan "Macau Terbuka GP Gold" adalah <organisasi>. Adanya tanda"-" membuat sistem melihat bahwa kalimat tersebut adalah satu kesatuan. Kesalahan pada sistem ini terjadi karena belum ada perlakuan terhadap tanda baca tertentu yang dianggap menjadi penghubung antar kata.

Kesalahan pemberian label nama juga terjadi pada kata <nama>Lolosnya Simon</nama> seharusnya hanya kata "Simon" saja yang diberikan entitas nama. Kesalahan ini terjadi pada fitur Word Shape. Dimana fitur ini melihat kata yang diawali huruf besar mendapat nilai lebih dibandingkan kata tanpa diawali huruf besar. Pada dokumen kata "Lolosnya" diawali dengan huruf besar "L", sehingga probabilitas "Lolosnya Simon" merupakan satu nama menjadi lebih besar dibandingkan kata "Simon" saja yang merupakan nama.

4. Kesimpulan dan Saran

Setelah melakukan pembuatan program dan pengujian terhadap setiap fitur maka diperoleh beberapa kesimpulan sebagai berikut:

1. Program implementasi Stanford NER untuk bahasa Indonesia dapat memudahkan peneliti untuk melakukan pemilihan dan perubahan fitur. Kemudahan ini dapat membantu pelatihan pemberian entitas.
2. Fitur dengan hasil akurasi tertinggi adalah fitur Current Word yang menghasilkan akurasi sebesar 59%.
3. Secara keseluruhan hasil akurasi tertinggi diperoleh pada pengujian kedua yaitu pengujian dengan menambahkan daftar nama, lokasi, organisasi, hari, bulan dan tahun.
4. Entitas yang gagal untuk diberi label adalah entitas organisasi. Kesalahan pemberian label dapat terjadi karena kata tidak berhasil diberi label atau kata diberi label organisasi namun salah.

Hasil dari pemberian entitas dan pemilihan fitur masih jauh dari akurasi yang diinginkan. Walaupun demikian hasil dari program ini sudah dapat banyak membantu bagi peneliti untuk melakukan pemilihan fitur yang ada. Penelitian lebih lanjut diharapkan dapat difokuskan pada kombinasi fitur sehingga diperoleh hasil pemberian label yang akurat. Selain itu perlu dilakukan beberapa perlakuan khusus terhadap beberapa kata khusus yang berlaku di Indonesia. Seperti kata majemuk, kata berulang dan lainnya.

Daftar Pustaka

[1] Arman, Arry Akhmad, *Teknologi Pemrosesan Bahasa Alami sebagai Teknologi Kunci untuk Meningkatkan Cara Interaksi antara Manusia dengan Mesin,* Seminar Ilmiah Dr. Arry Akhmad Arman (Departemen Teknik Elektro, Fakultas Teknologi Industri – ITB), pada acara Sidang Terbuka Institut Teknologi Bandung, 23 Agustus 2004.

[2] Brants, Thorsten, *Natural Language Processing in Information Retrieval,* In Proceedings of the 14th Meeting of Computational Linguistics in the Netherlands, p.1-13, 2004.
[3] Brill, Eric (1992), *A Simple Rule-Based Part of Speech Tagger*, ANLC '92 Proceedings of the third conference on Applied natural language processing, P.152-155.
[4] Chandrawati, Triastuti, *Pengembangan Part Of Speech Entitasger untuk Bahasa Indonesia Berdasarkan Metode Conditional Random Fields dan Transformation Based,* Fakultas Ilmu Komputer Universitas Indonesia (Skripsi tidak dipublikasikan), 2004
[5] Liu, X. and Croft, W. B., Statistical language modeling for information retrieval. Annu. Rev. Info. Sci. Tech., 39: 1–31, 2006
[6] Cutting, Doug. etc. *A Practical Part-of-Speech Tagger*, IN PROCEEDINGS OF THE THIRD CONFERENCE ON APPLIED NATURAL LANGUAGE PROCESSING, p.133-140, 1992.
[7] Lafferty, John, Andrew McCallum, and Fernando Pereira. *Conditional random fields: Probabilistic models for segmenting and labeling sequence data.* In Proceedings of ICML 2001.
[8] Christopher Manning and Dan Klein. 2003. *Optimization, Maxent Models, and Conditional Estimation without Magic.* Tutorial at HLT-NAACL 2003 and ACL 2003.
[9] Jenny Rose Finkel, et al., *Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling*, ACL '05 Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, pages 363-370, 2005.

Viny Christanti M., M.Kom, memperoleh gelar M.Kom dari Fakultas Ilmu Komputer, Universitas Indonesia. Saat ini aktif mengajar di Fakultas Teknologi Informasi, Universitas Tarumanagara.

Jeanny Pragantha, M.Eng., Dosen Teknik Informatika, Universitas Tarumanagara.

Andreas Aditya, S.Kom, memperoleh gelar S.Kom dari Fakultas Teknologi Informasi, Universitas Tarumanagara

SRITI2013

ORIGINALITY REPORT

17%	16%	6%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	bahasalami.blogspot.com Internet Source	1%
2	eprints.umm.ac.id Internet Source	1%
3	Submitted to Universitas Dian Nuswantoro Student Paper	1%
4	e-jurnalpenelitian.blogspot.com Internet Source	1%
5	ceur-ws.org Internet Source	1%
6	123dok.com Internet Source	1%
7	repository.maranatha.edu Internet Source	1%
8	Submitted to Hong Kong Baptist University Student Paper	1%
9	www.scribd.com Internet Source	1%

10	Submitted to University of New York in Tirana Student Paper	1%
11	Submitted to Universitas Brawijaya Student Paper	1%
12	sriti.akakom.ac.id Internet Source	1%
13	nbn-resolving.org Internet Source	1%
14	expertupdate.org Internet Source	1%
15	repository.its.ac.id Internet Source	1%
16	"Handbuch der Künstlichen Intelligenz", Walter de Gruyter GmbH, 2003 Publication	<1%
17	student.blog.dinus.ac.id Internet Source	<1%
18	eprints.mdp.ac.id Internet Source	<1%
19	Zhou, Dong, Mark Truran, Tim Brailsford, Vincent Wade, and Helen Ashman. "Translation techniques in cross-language information retrieval", ACM Computing Surveys, 2012. Publication	<1%

20	media.neliti.com Internet Source	<1%
21	Submitted to STIKOM Surabaya Student Paper	<1%
22	theses.gla.ac.uk Internet Source	<1%
23	jurnal.unimed.ac.id Internet Source	<1%
24	repository.unikom.ac.id Internet Source	<1%
25	jatp.ift.or.id Internet Source	<1%
26	eprints.dinus.ac.id Internet Source	<1%
27	authorzilla.com Internet Source	<1%
28	alta2017.alta.asn.au Internet Source	<1%
29	id.123dok.com Internet Source	<1%
30	pt.scribd.com Internet Source	<1%
31	widuri.raharja.info Internet Source	

		<1%
32	www.aamva.org Internet Source	<1%
33	agustinus-hardiyanto.blogspot.com Internet Source	<1%
34	eprints.uny.ac.id Internet Source	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		