PENGENALAN ENTITAS BERNAMA PADA TEKS BAHASA INDONESIA MENGGUNAKAN CONDITIONAL RANDOM FIELDS

Narti Jaariyah[1], Ednawati Rainarli, S.Si., M.Si.[2]

[1,2] Teknik Informatika, Teknik dan Ilmu Komputer, Universitas Komputer Indonesia
Jalan Dipatiukur No. 112-114, Coblong, Bandung, Jawa Barat 40132, Indonesia
E-mail : nartijaariyah@gmail.com[1], ednawati.rainarli@email.unikom.ac.id[2]

ABSTRAK

Pengenalan entitas bernama merupakan suatu proses untuk mengklasifikasi entitas nama seperti nama orang, lokasi, organisasi, waktu, dan kuantitas pada suatu teks. Pengenalan entitas bernama pada teks bahasa Indonesia sudah pernah dilakukan menggunakan metode Hidden Markov Model. Namun pada penelitian berikutnya dilakukan pengembangan terhadap metode HMM yakni metode Conditional Random Fields. CRF memiliki banyak kelebihan dibandingkan metode Hidden Markov Model dan Maximum Entropy Markov Model. Hal ini terbukti pada penerapan pengenalan entitas bernama menggunakan metode CRF pada berbagai bahasa yang menghasilkan nilai akurasi yang tinggi. Simulator pengenalan entitas bernama dibuat untuk menguji seberapa baik CRF dalam mengenali entitas bernama. Fitur yang digunakan adalah kelas kata sekarang, kelas kata sekarang dan kelas kata sebelumnya, dan kelas kata sekarang, kelas sebelumnya, dan kelas kata setelahnya. Penilaian simulator menggunakan *f-measure* dengan cara membandingkan label entitas hasil simulator dengan jenis entitas yang sebenarnya. Pengujian menggunakan data latih dan data uji yang sama hasil akurasi terbaik yang diperoleh sebesar 90.53% dengan recall 63.09% dan precision 31.55%. Hasil pengujian terhadap data latih dan data uji yang berbeda menunjukkan nilai akurasi terbaik adalah 90.06% dengan recall dan precision adalah 68.38% dan 41.35%.

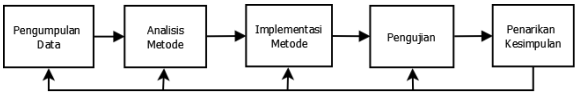
Kata kunci : ekstraksi informasi, pengenalan entitas bernama, conditional random fields, kelas kata

1. PENDAHULUAN

Pengenalan entitas bernama banyak digunakan dalam pemprosesan bahasa alami seperti ekstraksi informasi, mesin penjawab otomatis, peringkasan, temu-kembali informasi, mesin penerjemah, anotasi video, biofarmatik dan sebagainya. Sistem pengenalan entitas bernama untuk teks bahasa Indonesia sudah dilakukan menggunakan metode *Hidden Markov Model* [1]. Namun pada penelitian ekstraksi kalimat pertanyaan yang menggunaan sistem pengenalan entitas masih terdapat kegagalan yang disebabkan sistem pengenalan entitas masih belum akurat mengenali entitas bernama pada teks [2]. Selain metode HMM terdapat metode lain yang dapat digunakan untuk melakukan pengenalan entitas bernama yakni *conditional random fields*. Metode CRF untuk kasus pengenalan entitas bernama berhasil diimplementasikan pada berbagai bahasa di dunia dan menghasilkan nilai akurasi yang tinggi seperti pada bahasa Inggris dengan akurasi 92,29% dan bahasa Bengali nilai akurasi mencapai 90,7% [4,5]. Dari penelitian-penelitian yang telah dilakukan terlihat bahwa pengenalan entitas bernama menggunakan CRF memiliki nilai akurasi yang cukup tinggi. Namun, penelitian tersebut masih belum diaplikasikan pada bahasa Indonesia sedangkan setiap bahasa memiliki aturan-aturan penulisan bahasa tersendiri. Oleh karena itu masih diperlukan suatu penelitian untuk mengetahui keakuratan metode CRF dalam mengenali entitas bernama khususnya pada teks bahasa Indonesia.

2. ISI PENELITIAN

Metodologi yang digunakan dalam penelitian ini yaitu pengumpulan data, analisis metode, implementasi metode, pengujian, dan penarikan kesimpulan yang dapat dilihat pada Gambar 1.



Gambar 1 Tahap Penelitian

Dari gambar 1 tahapan-tahapan metodologi penelitian yang digunakan adalah sebagai berikut.

1. Pengumpulan Data

Metode pengumpulan data yang digunakan dalam penelitian ini yakni studi literatur. Tahap ini dilakukan dengan cara mempelajari literatur-literatur seperti buku, jurnal, artikel ilmiah, dan *website* yang berhubungan dengan pengenalan entitas bernama dan c*onditional random field*. Tujuan dari studi literatur ini adalah sebagai dasar teori pembangunan simulator pengenalan entitas bernama.

2. Analisis Metode

Pada tahapan ini dilakukan analisis kebutuhan dan proses-proses metode *conditional random fields* untuk mengenali entitas bernama. Setelah kebutuhan metode dan kebutuhan simulator terkumpul maka dibuat perancangan simulator yang akan dibangun.

3. Implementasi Metode

Proses yang dilakukan dalam tahapan ini adalah membangun simulator pengenalan entitas

berdasarkan analisis metode yang telah dilakukan. Simulator dibangun menggunakan bahasa pemograman Java dan teks editor sebagai tempat menyimpan data pelatihan dan pengujian.

4. Pengujian

Setelah simulator dibangun tahap selanjutnya adalah melakukan pengujian hasil implementasi *conditional random fields* untuk mengenali entitas bernama pada teks bahasa Indonesia. Pengujian ini dilakukan dengan berfokus pada pengujian akurasi data.

5. Penarikan kesimpulan

Kesimpulan didapatkan dari hasil implementasi metode *conditional random fields* untuk mengenali entitas bernama pada teks bahasa Indonesia berdasarkan hasil pengujian yang telah dilakukan pada tahap sebelumnya.

2.1. Data Masukkan

Data masukkan adalah teks bahasa Indonesia yang sesuai dengan aturan baku bahasa Indonesia. Teks terbagi atas 2 yaitu teks sebagai data pelatihan dan data pengujian. Data pelatihan terdiri 2.231 kalimat yang terdiri dari 43.301 kata dengan 1.756 kata nama orang, 1.546 kata nama lokasi, 1.417 kata termasuk dalam entitas organisasi, dan sisanya 38.582 kata termasuk dalam entitas o*ther*. Data pengujian terdiri dari 614 kalimat dengan 747 kata nama orang, 226 nama lokasi, 444 nama organisasi dan bukan entitas sebanyak 9.669 kata.

2.2. Praproses

Praproses merupakan tahap awal untuk melakukan pengenalan entitas bernama pada teks. Tahapan dalam praproses terdiri dari :

a. Pemisahan kalimat

Pemisahan kalimat dilakukan menggunakan aturan-aturan pada bahasa Indonesia seperti diakhiri dengan tanda titik(.), tanda seru (!) dan tanda tanya (?), serta di dalamnya bisa terdapat beberapa baca yaitu tanda koma (,), tanda titik koma (;), tanda hubung (-), tanda pisah (—), tanda elipsis (…), tanda kurung ((…)), tanda kurung siku ([…]), tanda petik ("..."), tanda petik tunggal ('…') tanda garis miring (/) dan tanda penyingkat atau apostrof [9]. Dalam proses ini akan dilakukan menggunakan *regular expression* dan diberi tag <START> untuk awal kalimat dan <END> untuk akhir kalimat.

b. Tokenisasi

Tokenisasi merupakan proses pemisahan kalimat-kalimat menjadi kumpulan kata-kata yang sesuai dengan urutannya. Tanda baca dalam teks dianggap sebagai sebuah kata.

c. Pengenalan Kelas Kata

Setelah ditokenisasi, setiap kata dilakukan proses pengenalan kelas kata menggunakan POS *Tagging* untuk membantu pelabelan jenis kata. POS *Tagging* yang digunakan adalah IPOSTagger hasil penelitian Alfan Wicaksono untuk kategori teks bahasa Indonesia menggunakan *Hidden Markov Model*.

2.2 Ekstraksi Fitur

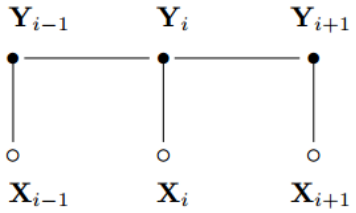
Pada implementasinya, fungsi fitur adalah sebuah fungsi yang mengkombinasikan sebuah fitur dengan labelnya yang bersesuaian dan antara label yang berdekatan. Contoh fungsi fitur dapat dilihat pada persaaan 3.

$$f_{1000}(y_{10}, x, 10) = \begin{cases} 1, & if\ x_{10} = "today"\ and\ y_{10} = B - NP \\ 0, & otherwise \end{cases} \quad (3)$$

Dimana, fungsi fitur ke-1000 akan menghasilkan nilai 1 bila kata ke-10 adalah "today" dan label saat ini adalah B-NP dan akan bernilai 0 untuk label lain.

2.3 Conditional Random Fields

Conditional random fields merupakan suatu model probabilistik untuk segmentasi dan pelabelan suatu sekuen data [6]. CRF memiliki banyak kelebihan dibandingkan model probabilitas lain seperti *Hidden Markov Model* (HMM) dan *Maximum Entropy Markov Model* (MEMM). CRF mengatasi permasalahan ketergantungan asumsi yang tinggi pada HMM. Hal ini karena CRF dapat menentukan sendiri seberapa banyak fitur yang diinginkan untuk membangun sebuah model CRF tidak seperti HMM yang bersifat lokal dimana setiap kata hanya bergantung pada label saat ini dan setiap label sebelumnya. Selain itu CRF dapat mempunyai bobot yang bebas sedangkan HMM harus memenuhi bobot tertentu. CRF juga mengatasi permasalahan label bias pada MEMM karena CRF memformulasikan distribusi kondisional label untuk sekuens data secara kesuluruhan dibandingkan MEMM yang memformulasikan distribusi kondisional label untuk setiap elemen data [7]. CRF digunakan pada banyak aplikasi termasuk *natural language processing, computer vision*, dan *bioinformatic*. Salah satu bentuk dari CRF adalah linier-chain CRF yang ditunjukkan oleh Gambar 2.



Gambar 2 *Linear Chain CRF*

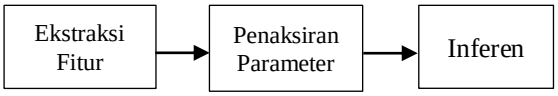
Diberikan dua peubah acak dimana $\bar{x} = (x_1, x_2, \dots, x_n)$ adalah himpunan data observasi dan $\bar{y} = (y_1, y_2, \dots, y_n)$ adalah himpunan label yang mungkin. Probabilitas kondisional dari y terhadap x dapat dituliskan dalam persamaan 1.

$$p(y|x) = \frac{1}{z(x)} \prod_c \psi_c(y_c, x) \quad (1)$$

Dimana $\psi_c(y_c, x)$ adalah fungsi positif yang selanjutnya disebut fungsi potensial dan z(x) menunjukkan fungsi normalisasi dari distribusi probabilitas kondisional label untuk semua sekuens data x dan dinotasikan dengan persamaan 2.

$$Z(x) = \sum_y \prod_c \psi_c(y_c, x) \quad (2)$$

Fungsi potensial mengacu pada jumlah yang mengikat label data dengan fitur pada waktu yang sama. Fitur biasanya disebut juga sebagai data pattern pada beberapa literatur CRF. Dalam kasus yang sederhana fungsi potensial merupakan eksponensial dari jumlah bobot semua fungsi fitur. Secara umum proses CRF ditunjukkan pada Gambar 3 berikut ini.



Gambar 3 Tahapan proses CRF

2.3.1 Penaksiran Parameter

Penaksiran parameter adalah proses untuk mendapatkan nilai optimal parameter fungsi fitur yakni λ_k. Nilai λ_k dihitung mengunakan prosedur maksimum *likelihood*. Maksimum *likelihood* merupakan kuantitas yang menunjukkan berapa banyak parameter yang didukung oleh data pelatihan. Maksimum *likelihood* dapat memaksimalkan kemiripan ciri dari kumpulan data pelatihan yang telah dimodelkan dengan CRF dengan kumpulan data pelatihan. Metode ini membutuhkan dua perhitungan yaitu *log-likelihood* dan turunan pertamanya [14]. *Log-likelihood* secara matematis dapat dituliskan dalam persamaan 4 berikut.

$$\mathcal{L} = \sum_{t\epsilon[1,T]} \sum_k \lambda_k f_k(y'_t, x, t) - \sum_{t\epsilon[1,T-1]} \sum_k \lambda_k f_k(y'_t, y'_{t+1} x, t) - \log Z(x) - \sum_k \frac{\lambda_k^2}{2\sigma^2} \quad (4)$$

Dari persamaan 4 maka didapatkan turunan dari *log-likelihood feature function* ke-*k* pada persamaan 5 dan 6.

$$G_k^x = \sum_{t\epsilon[1,T]} (f_k(y_t', x, t) - \sum_{y_t} P_t(y_t|x) f_k(y_t, x, t)) - \frac{\lambda_k}{\sigma^2} \quad (5)$$

$$G_{k\prime}^x = \sum_{t\epsilon[1,T-1]} (f_k(y'_t, y'_{t+1} x, t) - \sum_{y_t} \sum_{y_{t+1}} P_t(y_t, y_{t+1}|x) f_k(y_t, y_{t+1} x, t) - \frac{\lambda_{k\prime}}{\sigma^2} \quad (6)$$

Keterangan :

G_k^x : nilai gradien ke-k untuk fungsi fitur node
$G_{k\prime}^x$: nilai gradien ke-k untuk fungsi fitur edge
k : bernilai 1 sampai banyaknya fitur node
k' : bernilai 1 sampai banyaknya fitur edge
T : banyaknya kata dalam 1 kalimat
$P_t(y_t|x)$: probabilitas lokal node ke-t
$P_t(y_t, y_{t+1}|x)$: probabilitas lokal edge ke-t
$f_k(y'_t, x, t)$: fungsi fitur node ke-k dari data pelatihan yang telah diberi label
$f_k(y_t, x, t)$: fungsi fitur ke-k untuk semua label y_t yang mungkin
$f_k(y_t, y_{t+1} x, t)$: fungsi fitur ke-k untuk semua label y_t dan y_{t+1} yang mungkin
$f_k(y'_t, y'_{t+1} x, t)$: fungsi fitur edge ke-k dari data pelatihan yang telah diberi label
σ : standar deviasi distribusi Gauss kisaran nilai 0.1 sampai 10 [8]
λ_k : nilai parameter fungsi fitur node ke-k
$\lambda_{k\prime}$: nilai parameter fungsi fitur edge ke-k

Untuk mendapatkan nilai parameter fungsi fitur λ_k yang optimal, persamaan 5 dan 6 digunakan pemrograman dinamis dengan memanfaatkan prosedur *forward-backward pass* untuk menelusuri sekuens data. Penelusuran dengan *forward-backward pass* bertujuan untuk mendapatakan nilai dari seluruh probabilitas lokal label pada sekuens data. *Forward pass* pada tiap data x_t secara matematis pada persamaan 7 dan 8.

Untuk t=1

$$\alpha_1[y_1] = 1/S \quad (7)$$

Untuk t=2 sampai T

$$\alpha_t[y_t] = \kappa_t \sum_{y_{t-1}} \alpha_{t-1}[y_{t-1}] \, \phi_t(y_t, x) \, \psi_t(y_t, y_{t+1}, x) \quad (8)$$

Keterangan :

S = jumlah label
κ_t = faktor penskalaan sehingga $\sum_{y_t} \alpha_t[y_t] = 1$, $\kappa_t > 0$
$\alpha_{t-1}[y_{t-1}]$ = nilai forward pass ke t-1
$\phi_t(y_t, x)$ = nilai node potensial ke-t
$\psi_t(y_t, y_{t+1}, x)$ = nilai edge potensial ke-t

B*ackward pass* pada tiap data x_t secara matematis dapat dituliskan dala persamaan 9 dan 10.

Untuk t=T

$$\alpha_1[y_1] = 1/S \quad (9)$$

Untuk t=2 sampai T-1

$$\beta_t[y_t] = \gamma_t \sum_{y_{t+1}} \beta_{t+1}[y_{t+1}] \phi_{t+1}(y_{t+1}, x) \, \psi_t(y_t, y_{t+1}, x) \quad (10)$$

Keterangan :

γ_t = faktor penskalaan sehingga $\sum_{x_t} \beta_t[x_t] = 1$, $\gamma_t > 0$
$\beta_{t+1}[y_{t+1}]$ = nilai backward pass ke-t+1
$\phi_{t+1}(y_t, x)$ = nilai node potensial ke-t+1
$\psi_t(y_t, y_{t+1}, x)$ = nilai edge potensial ke-t

Setelah semua *forward pass* dan *backward pass* untuk tiap *data* x telah memiliki nilai, maka perhitungan probabilitas lokal untuk setiap label yang mungkin pada data x_t dapat dituliskan dalam persamaan matematis 11 dan 12 sebagai berikut.

$$P_t(y_t|x) = w_t \alpha_t[y_t] \phi_t(y_t, x) \beta_t[y_t] \quad (11)$$

$$P_t(y_t, y_{t+1}|x) = \text{Ш}_t \alpha_t[y_t] \phi_t(y_t, x) \psi_t(y_t, y_{t+1}, x) \phi_{t+1}(y_t, x) \beta_{t+1}[y_{t+1}] \quad (12)$$

Keterangan :

$P_t(y_t|x)$: probabilitas lokal untuk setiap label y_t yang mungkin
w_t : faktor normalisasi untuk memastikan bahwa $\sum_{y_t} P_t(y_t|x) = 1$.
$\alpha_t[y_t]$: nilai *forward pass* ke-t
$\phi_t(y_t, x)$: nilai node potensial ke-t
$\psi_t(y_t, y_{t+1}, x)$: nilai edge potensial ke-t
$\beta_t[y_t]$: nilai *backward pass* ke-t
$P_t(y_t, y_{t+1}|x)$: probabilitas lokal untuk setiap label $y_t$dan y_{t+1} yang mungkin
W_t : faktor normalisasi memastikan $\sum_{y_t} \sum_{y_{t+1}} P_t(y_t, y_{t+1}|x) = 1$
$\phi_{t+1}(y_t, x)$: nilai node potensial ke-t+1
$\beta_{t+1}[y_{t+1}]$: nilai *backward pass* ke-t+1

Setelah nilai probabilitas lokal tiap label yang mungkin untuk tiap data x_t pada sekuens *x* didapatkan, maka nilai turunan pertama *log-likelihood* akan didapatkan berdasarkan persamaan 5 dan 6. Kemudian nilai turunan pertama *log-likelihood* G_k^x akan digunakan untuk memperbaharui parameter fungsi fitur λ_k pada tiap iterasi berdasarkan persamaan 13 dan 14.

$$\lambda_k \leftarrow \lambda_k + \omega G_k^x \quad (13)$$

$$\lambda_k \leftarrow \lambda_k + \omega G_{k'}^x \quad (14)$$

Keterangan :
ω : *learning rate* dengan $\omega \epsilon$ [0.001,0.1] [8].
G_k^x : nilai gradien node
$G_{k'}^x$: nilai gradien edge

2.3.2 Inferen

Proses berikutnya setelah mendapatkan nilai parameter fitur dari proses pelatihan yakni mengimplementasikan nilai parameter tersebut pada data pengujian.Himpunan parameter fungsi fitur yang didapatkan dari proses pelatihan akan digunakan untuk melakukan inferen terhadap sekumpulan data uji yang akan diprediksi semua labelnya dengan menemukan sekuens label yang paling optimal secara keseluruhan. Ada tiga tahap untuk melakukan *decoding* yaitu menghitung fungsi potensial, *maximal forward pass*, dan *backtracking*.

Pada awal iterasi, $\alpha_1^{max}[x_1]$ diinisialisasi dengan nilai $1/S$ untuk tiap label dengan *S* adalah jumlah label yang mungkin diberikan pada sekuens data uji. Selanjutnya, penelusuran *maximal forward pass* dilakukan terhadap sekuens data secara matematis dapat dituliskan dalam persamaan 14 dan 15:

Untuk t=T

$$\alpha_1[y_1] = 1/S \quad (14)$$

Untuk t=2 sampai T-1

$$\alpha_t^{max}[y_t] = \kappa_t max_{yt-1}(\alpha_{t-1}[y_{t-1}^{max}]\phi_t(y_t, x)\psi_t(y_t, y_{t+1}, x)) \quad (15)$$

Keterangan :
S : jumlah label
κ_t : faktor normalisasi
$\alpha_{t-1}[y_{t-1}^{max}]$: nilai maksimal *forward pass* t-1
$\phi_t(y_t, x)$: nilai node potensial ke-t
$\psi_t(y_t, y_{t+1}, x)$: nilai edge potensial ke-t

Pada tahap selanjutnya yaitu *Backtracking*, bertujuan untuk menelusuri kembali dan meninjau label optimal pada tiap data berdasarkan nilai dari maximal forward pass yang dilakukan sebelumnya. Perhitungan *backtracking* secara matematis dapat ditulis dalam persamaan 16.

$$y_t^* = arg\ max_{y_t}(a_t^{max}[y_t]\ \phi_t(y_t, z)) \quad (16)$$

sehingga sekuens label optimal adalah $\{y_1^*, y_2^*, \dots, y_T^*\}$ untuk sekuens data pengujian sejumlah *T*.

Dimana :
$a_t^{max}[y_t]$ = nilai maksimal forward pass ke-t
$\phi_t(y_t, z)$ = nilai node potensial ke-t

2.4 Pengujian

Penelitian ini menggunakan metode pengujian *precision*, *recall*, dan *f-measure*. Perhitungan precision, recall, dan *f-measure* adalah sebagai berikut [10].

$$Precision = \frac{true\ positif}{\text{true positif+ false positif}} \quad (17)$$

$$Recall = \frac{\text{true positif}}{\text{true positif + false negatif}} \quad (18)$$

$$F - Measure = \frac{true\ positif + true\ negatif}{true\ positif + false\ positif + true\ negatif + false\ negatif} \quad (19)$$

Untuk entitas *person*, *location*, dan *oganization* yang diidentifikasi secara benar dihitung sebagai *true positif*, sedangkan yang salah dihitung sebagai *false positif*. Sedangkan untuk entitas *other* yang diidentifikasikan secara benar dihitung sebagai *true negatif*, sedangkan yang salah dihitung sebagai *false negatif*.

2.5 Skenario Pengujian

Pengujian dilakukan dengan cara mencari nilai parameter yang optimal dengan mengganti-ganti kombinasi parameter yang dimasukkan. Rencana pengujian yang dilakukan adalah sebagai berikut :

a. Meneliti pengaruh nilai standar deviasi dan *learning rate* menggunakan data latih dan data berbeda yang sama terhadap fitur kelas kata sekarang (fitur 1) , fitur kelas kata sekarang dan kelas kata sebelumnya (fitur 2), dan fitur kelas kata sekarang, kelas kata sebelumnya, dan kelas kata setelahnya (fitur 3)
b. Meneliti pengaruh nilai standar deviasi dan *learning rate* menggunakan data latih dan data uji yang berbeda terhadap fitur 1, 2, dan

2.6 Evaluasi Pengujian

Pengujian menggunakan standar deviasi pada rentang nilai 0.1 sampai dengan 10 dan *learning rate* berada pada rentang nilai 0.001 sampai dengan 0.1 [8]. Nilai standar deviasi Gauss diujikan dari nilai 0.1 dengan perubahan nilai kelipatan 3. Sedangkan

untuk parameter *learning rate* pengujian diberikan nilai *learning rate* 0,1 terlebih dahulu. Setelah itu, dilakukan percobaan dengan kelipatan 10^{-1} untuk melihat nilai *learning rate* yang akan menghasilkan nilai yang paling baik. Hasil pengujian menggunakan data latih dan data uji yang berbeda terhadap fitur 1, fitur 2, dan fitur 3 dapat dilihat pada Tabel 1,2,dan 3.

Tabel 1 Hasil Pengujian Akurasi Fitur 1

σ	ω	**P**	**R**	**F**
0.1	0.0001	0	0	87.04
	0.001	0.35	19.23	87.07
	0.01	0	0	84.92
	0.1	0	0	0
3	0.0001	42.7	61.86	89.31
	0.001	58.86	21.21	66.8
	0.01	24.28	10	62.39
	0.1	12.7	40	86.41
6	0.0001	42.7	61.86	89.31
	0.001	58.86	21.21	66.8
	0.01	58.86	21.32	66.99
	0.1	6.77	26.23	85.65
9	0.0001	42.7	61.86	89.31
	0.001	58.86	21.21	66.8
	0.01	58.86	21.32	66.99
	0.1	6.77	26.23	85.65
12	0.0001	42.7	61.86	89.31
	0.001	58.86	21.21	66.8
	0.01	58.86	21.32	66.99
	0.1	41.35	68.38	90.06

Tabel 2 Hasil Pengujian Akurasi Fitur 2

σ	ω	**P**	**R**	**F**
0.1	0.0001	0.35	18.52	87.06
	0.001	0.35	14.71	87
	0.01	0	0	82.71
	0.1	0	0	0
3	0.0001	4.59	30.66	86.48
	0.001	32.46	26.65	79.95
	0.01	32.46	26.65	79.95
	0.1	21.74	19.36	78.42
6	0.0001	4.59	30.66	86.48
	0.001	32.46	26.65	79.95
	0.01	43.9	32.55	81.2
	0.1	32.46	26.54	79.88
9	0.0001	4.59	30.66	86.48
	0.001	32.46	26.65	79.95
	0.01	43.9	32.55	81.2
	0.1	32.46	26.45	79.83
12	0.0001	4.59	30.66	86.48
	0.001	32.46	26.65	79.95
	0.01	43.9	32.55	81.2
	0.1	43.9	32.33	81.08

Tabel 3 Hasil Pengujian Akurasi Fitur 3

σ	ω	**P**	**R**	**F**
0.1	0.0001	1.55	30.56	86.97
	0.001	0.21	1.47	85.43
	0.01	0	0	82.55
	0.1	0	0	0
3	0.0001	4.52	29.91	86.44
	0.001	4.59	29.95	86.43
	0.01	4.59	29.55	86.41
	0.1	7.55	24.94	85.28
6	0.0001	4.52	29.91	86.44
	0.001	4.59	29.95	86.43
	0.01	12.42	47.44	87.05
	0.1	3.46	17.56	85.59
9	0.0001	4.52	29.91	86.44
	0.001	4.52	29.63	86.42
	0.01	12.42	47.44	87.05
	0.1	4.59	19.82	85.43
12	0.0001	4.52	29.91	86.44
	0.001	4.52	29.63	86.42
	0.01	12.42	47.44	87.05
	0.1	8.26	29.32	85.73

σ = Deviasi, ω = Learning Rate, P = Precision, R=Recall, F = Akurasi, TN = True Negative, FN= False Negative, TP=True Positive, FP=False Positive

Implementasi metode *conditional random fields* dalam mengenali entitas bernama pada teks bahasa Indonesia didapatkan hasil akurasi terbaik menggunakan data pelatihan dan data pengujian yang sama pada fitur 1 dengan deviasi 12 dan *learning rate* 0.1 adalah 90.53% yang dapat mengenali entitas nama orang, lokasi, dan organisasi sebanyak 1.489 kata dari 4.719 kata , untuk fitur 2 entitas yang dapat dikenali sebanyak 137 kata dari 4.582 kata entitas dengan akurasi sebesar 89.71% dengan deviasi 0.1 dan *learning rate* 0.0001, dan fitur 3 akurasi yang diperoleh mencapai 89.13% dengan entitas yang dapat dikenali sebanyak 103 kata dari 4.616 kata deviasi 0.1 dan *learning rate* 0.0001. Dari hasil ini dapat dilihat pada fitur 2 dan 3 memiliki nilai akurasi terbaik meskipun hanya mengenali sedikit kata entitas dibandingkan fitur 1. Hal ini karena fitur 2 dan 3 mengenali kata bukan entitas yang salah diprediksi masing-masing adalah 108 kata dan 91 kata dengan pemilihan deviasi dan *learning rate* yang sama yakni 0.1 dan 0.0001.

Hasil akurasi terbaik fitur 1 menggunakan data pelatihan dan pengujian yang berbeda diperoleh 90.06% dengan deviasi 12 dan *learning rate* 0.1 sehingga dapat dikenali 586 kata entitas dari 1.417 kata entitas. Untuk fitur 2 dikenali 5 kata entitas dan 9.647 kata bukan entitas sehingga akurasi terbaik yang diperoleh adalah 87.06% dengan deviasi dan learning rate yang dipilih adalah 0.1 dan 0.0001. Fitur 3 memiliki nilai akurasi terbaik saat nilai deviasi 6 ,9, 12 dan *learing rate* 0.01 sebesar

87.05% dengan entitas kata yang dikenali sebesar 116 dari 1.241 kata entitas.

3. PENUTUP

Dari penelitian yang telah dilakukan dapat disimpulkan simulator pengenalan entitas bernama pada teks bahasa Indonesia menggunakan metode *conditional random fields* dapat dilakukan menggunakan fitur kelas kata. Nilai akurasi terbaik untuk setiap difitur menggunakan data latih dan data uji yang sama maupun yang berbeda yaitu:

a. Nilai akurasi terbaik menggunakan data latih dan data uji yang sama untuk fitur kelas kata sekarang adalah 90.53%, fitur kelas kata sekarang dan kelas kata sebelumnya adalah 89.17%, dan fitur kelas kata sekarang, kelas kata sebelumnya dan kelas kata setelahnya adalah 89.13% .
b. Nilai akurasi terbaik menggunakan data latih dan data uji yang berbeda untuk fitur kelas kata sekarang adalah 90.06% , fitur kelas kata sekarang dan kelas kata sebelumnya adalah 87.06%, dan fitur kelas kata sekarang, kelas kata sebelumnya dan kelas kata setelahnya 87.05% .

Saran untuk penelitian berikutnya adalah sebagai berikut.

a. Penelitian berikutnya dapat menggunakan fitur bentuk kata (*word shape*), n-gram, morfologi, nonmorfologi, atau menggunakan fitur yang lain untuk meningkatkan nilai akurasi.
b. Penelitian berikutnya dapat menggunakan algoritma *Baum-Weltch* untuk mengatur model yang didapatkan dari proses pelatihan menjadi maksimal.

DAFTAR PUSTAKA

[1] A. F. Wicaksono and A. Purwarianti, "HMM Based Part-of-Speech Tagger for Bahasa Indonesia," in *Proceeding of the 4th-International MALINDO Workshop (MALINDO2010)*, 2010.

[2] Y. Syaifudin, “Identifikasi Kalimat Kutipan Dari Teks Berita Online Berbahasa Indonesia Dengan Metode Berbasis Aturan,“ Skripsi, Ilmu Komputer, UGM, Yogyakarta, 2016.

[4] J. R. Finkel, T. Grenager and C. D. Manning, "Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, 2005.

[5] A. Ekbal, R. Haque and S. Bandyopadhyay, "Named Entity Recognition in Bengali: A Conditional Random Field Approach," in *IJCNLP*, 2008.

[6] C. Sutton and A. McCallum, "An Introduction to Conditional Random Fields," *Machine Learning*, vol. 4, no. 4, pp. 267-373, 2011.

[7] H. M. Wallach, "Conditional Random Fields: An Introduction," Technical Reports (CIS), 2004.

[8] T. T. Truyen and P. Dinh, "A Practitioner Guide to Conditional Random Fields for Sequential Labelling," Curtion University of Technology, 2008.

[9] “Kemendikbud, n.d., Pedoman Umum Ejaan Bahasa Indonesia,” [Online] Available : http://badanbahasa.kemdikbud.go.id/lamanbahasa/sites/default/files/pedoman_umum-ejaan_yang_disempurnakan.pdf, [Accesed 3 Januari 2017].

[10] E. Prasetyo, Mengolah Data Menjadi Informasi Menggunakan Matlab, Yogyakarta: ANDI, 2014.