# *Part of Speech Tagger* untuk Bahasa Indonesia Menggunakan *Conditional Random Field* (CRF)

Fivip Saefulloh

Teknik Informatika – Universitas Komputer Indonesia
Jl. Dipatiukur 112-114 Bandung
Email : fvpsaefulloh@gmail.com

**ABSTRAK**

*Part-of-speech tagging* merupakan proses penandaan untuk menentukan kelas kata untuk setiap kata dalam teks input [1]. Pada penelitian ini metode yang digunakan untuk membangun POS *Tagger* adalah *Conditional Random Field* (CRF). CRF merupakan salah satu metode probabilistik yang digunakan dalam pengembangan POS *Tagging* dan memiliki kelebihan diantara metode probabilistik lain, seperti *Hidden Markov Model* (HMM) serta Maximum Entropy Methods [8,9]. Untuk data *training* dan data *testing* yang digunakan berasal dari "Indonesian_Manually_Tagged_Corpus" [2] berisi 250000 kata dengan 23 tagset. Data *training* yang digunakan terdiri dari 3910 kata dan untuk testing digunakan 4 file data *testing* yang terdiri dari 50 kalimat dengan jumlah kata yang berbeda. Dari hasil pengujian yang dilakukan rata-rata akurasi yang dihasilkan adalah 78,20%.

**Kata Kunci** : Pelabelan kelas kata, CRF, Bahasa Indonesia, kelas kata

## 1. PENDAHULUAN

Dalam Bahasa Indonesia sudah dilakukan beberapa penelitian mengenai POS *tagging*, yaitu: *Probabilistic Part Of Speech Tagging for Bahasa Indonesia* [3] yang menggunakan metode probabilistic *Maximum Entropy Models* dengan nilai akurasi tertinggi mencapai 97,57% serta *Conditional Random Field* dengan nilai akurasi tertinggi mencapai 91,15%, *HMM Based Part-of-Speech Tagger for Bahasa Indonesia* [4] dengan akurasi 96,50% untuk teks yang diambil sebagian (15%) dari luar korpus dan 80,4% untuk teks yang berada di luar korpus, *Pengembangan Part of Speech Tagger untuk Bahasa Indonesia Berdasarkan Metode Conditional Random Fields dan Transformation Based Learning* [5] yang menghasilkan akurasi 80,21 % untuk metode CRF dan *Building an Indonesian Rule-Based Part-of-Speech Tagger* [6] dengan akurasi tertinggi 79%.

Secara umum, algoritma atau metode yang digunakan dalam pengembangan POS *Tagging* dibagi ke dalam tiga cara, yaitu [1,7] : *Rule-based Tagger*, *Stochastic Tagger* dan *Transformation-based Tagger*.

*Rule-based tagger* dilakukan dengan cara top-down, yaitu melakukan konsultasi dengan ahli linguistik untuk menentukan aturan-aturan yang biasa digunakan manusia.

*Stochastic tagger* dilakukan dengan cara *bottom-up*, yaitu dengan menggunakan korpus sebagai training data untuk menentukan secara probabilistik tag terbaik untuk sebuah kata dalam sebuah konteks.

*Transformation-based tagger* merupakan gabungan kedua cara diatas. Sama halnya seperti *halnya rule-based tagger* cara ini berdasarkan pada aturan-aturan yang sudah ditentukan ketika menemui kata-kata yang ambigu. Cara ini juga mempunyai komponen machine learning untuk mengambil aturan (rule) secara otomatis dari training korpus.

Adapun pada penelitian ini metode yang digunakan adalah *Stochastic tagger* dengan menggunakan *Conditional Random Field* (CRF). Pada bahasa Indonesia sendiri sudah dilakukan penelitian mengenai pengembangan metode ini dengan menghasilkan akurasi 91,15% dan 80,21% [4,6] pada penelitian tersebut menggunakan 25 dan 37 *tagset* dengan menggunakan korpus sebanyak 26346 kata. Dari hasil penelitian tersebut, akurasi yang dihasilkan masih belum maksimal. Berdasarkan pada hal tersebut, maka masih diperlukan penelitian lebih lanjut untuk meningkatkan akurasi dengan menggunakan metode ini. Pada penelitian ini korpus yang digunakan adalah "Indonesian_Manually_Tagged_Corpus" [2] berisi 250000 kata dengan 23 tagset.

### 1.1 Kata

Menurut Kamus Besar Bahasa Indonesia (KBBI), kata merupakan unsur bahasa yang diucapkan atau dituliskan yang merupakan perwujudan kesatuan perasaan dan pikiran yang dapat digunakan dalam berbahasa.

### 1.2 Kelas Kata

Dalam menentukan kelas kata, prinsip yang perlu dipegang ialah kenyataan bahwa kelas kata atau

kategori kata adalah bagian dari sintaksis, jadi ciri-ciri tiap kata harus dijelaskan dari sudut sintaksis. Berikut ini adalah pembagian kelas kata dalam bahasa Indonesia [14] :

Tabel 1 Kelas Kata Bahasa Indonesia

| No | Kelas Kata |
|---|---|
| 1 | Kata Kerja (Verba) |
| 2 | Kata Sifat (Adjektiva) |
| 3 | Kata Benda (Nomina) |
| 4 | Kata Ganti (Pronomina) |
| 5 | Kata Bilangan (Numeralia) |
| 6 | Kata Keterangan (Adverbia) |
| 7 | Kata Tanya (Interogativa) |
| 8 | Kata Tunjuk (Demonstrativa) |
| 9 | Kata Sandang / Sebutan (Artikula) |
| 10 | Kata Depan (Preposisi) |
| 11 | Kata Penghubung (Konjungsi) |
| 12 | Kategori Fatis |
| 13 | Kata Seru (Interjeksi) |

### 1.3 Conditional Random Field (CRF)

CRF merupakan *framework* untuk membangun model probabilistik untuk segmentasi dan pelabelan data yang berurutan. Kelebihan utama CRF dibandingkan dengan *hidden markov model* (HMM) adalah sifat bersyaratnya (*conditional*), sehingga mengurangi ketergantungan asumsi yang dibutuhkan oleh HMM untuk memastikan inferensi mudah dikerjakan [8]. Selain itu CRF juga menghindari masalah label bias, kelemahan yang ditunjukkan oleh *maximum entropy markov model* (MEMM) dan model *conditional markov* lain yang merupakan *directed graphical models* [9].

Diperkenalkan oleh Lafferty et al (2001) [8], *Conditional Random Field* (CRF) merupakan model probabilistik untuk menghitung probabilitas $p(y|x)$ dari output yang memungkinkan $y = (y_1, y_2, \dots, y_n)$ terhadap input yang diberikan $x = (x_1, x_2, \dots, x_n)$ yang disebut juga data observasi [10]. Probabilitas kondisional dari $y$ terhadap $x$ dapat dituliskan dalam persamaan [11] :

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

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

Dimana $\psi_c(y_{c,}x)$ merupakan fungsi positif atau fungsi potensial, dan $Z(x)$ merupakan fungsi partisi atau fungsi normalisasi. Dalam implementasinya, ada dua jenis fungsi potensial, yaitu *node* potensial $\phi(y_t, x)$ dan *edge* potensial $\psi(y_t, y_{t+1}, x)$. *Node* potensial dapat dinotasikan kedalam persamaan berikut [11].

$$\phi_t(y_t, x) = \exp(\sum_k \lambda_k f_k(y_t, x, t)) \quad (3)$$

Sedangkan untuk *edge* potensial dapat dituliskan sebagai berikut [11].

$$\psi_t(y_t, y_{t+1}, x) = \exp(\sum_k \lambda_{k'} f_{k'}(y_t, y_{t+1}, x, t)) \quad (4)$$

Dimana :

| | |
|---|---|
| $x$ | : data masukan |
| $y_t$ | : label untuk data $x$ |
| $y_{t+1}$ | : label setelah $y_t$ |
| $t$ | : indeks kata ke-t dari data input |
| $\lambda_k$ | : parameter fitur *node* |
| $\lambda_{k'}$ | : parameter fitur *edge* |
| $f_k(y_t, x, t)$ | : fitur *node* ke-k |
| $f_{k'}(y_t, y_{t+1}, t)$ | : fitur *edge* ke-k |

Secara umum tahapan yang dilakukan dalam metode *conditional random field* (CRF) adalah sebagai berikut.

#### 1.3.1 Ekstraksi Fitur

Fitur merupakan elemen paling penting untuk membangun model ini. Pemilihan fitur yang tepat dapat meningkatkan akurasi pelabelan. Dalam penelitian ini sendiri terdapat dua fitur, yaitu fitur *node* dan fitur *edge*. Fitur *node* merupakan asosiasi antara label *y* dengan data masukan $x$, berikut ini merupakan contoh fitur *node*, yaitu :

$$f_t(y_t, x, t) \begin{cases} 1 \; if \; y_t = CC \quad , x = Saya \\ 0 \quad otherwise \end{cases}$$

Fitur di atas merupakan fitur ke-t yang bernilai 1 jika label CC dan kata yang sedang diproses adalah saya, jika bukan maka bernilai 0.

Sedangkan fitur *edge* merupakan asosiasi antara label kelas kata untuk kata yang sedang diproses dengan label kelas kata untuk kata setelahnya. Berikut ini adalah contoh fitur *edge*, yaitu :

$$f_t(y_t, y_{t+1}, x, t) \begin{cases} 1 \; if \; y_t = NNP \; , y_{t+1} = VB \\ 0 \quad otherwise \end{cases}$$

Fitur di atas merupakan fitur ke-t yang bernilai 1 jika label kelas kata untuk kata yang sedang diproses merupakan *NNP* dan label kelas kata untuk kata setelahnya adalah *VB*, jika bukan maka bernilai 0.

#### *1.3.2 Training*

Proses *training* dilakukan untuk mendapatkan nilai optimal untuk parameter fitur *node* dan parameter fitur *edge*. Nilai parameter didapatkan dengan menggunakan *maximum likelihood* yang dapat dituliskan [11] :

$$\mathcal{L} = \sum_{t\in[1,T]}\sum_k \lambda_k f_k(y_t, x, t) - \sum_{t\in[1,T-1]}\sum_{k'} \lambda_{k'} f_{k'}(y_t, y_{t+1}, x, t) - \log Z(x) - \sum_k \frac{\lambda_k^2}{2\sigma^2} \quad (5)$$

Dimana :

| | |
|---|---|
| $t \in [1, T]$ | : banyaknya kata |
| $\lambda_k$ , $\lambda_{k'}$ | : parameter fitur ke-k |
| $f_k(y_t, x, t)$, $f_{k'}(y_t, y_{t+1}, x, t)$ | : fitur ke-k |
| $Z(x)$ | : fungsi partisi |
| $\sigma$ | : standar deviasi |

Untuk mendapatkan parameter λ untuk tiap fungsi fitur, maka digunakan turunan *log-likelihood*, yaitu [11] :

$$G_k^x = \sum_{t\in[1,T]}\left(f_k(y_t, x, t) - \sum_{y_t} P_t(y_t|x) f_k(y_t, x, t)\right) - \frac{\lambda_k}{\sigma^2} \quad (6)$$

$$G_{k'}^x = \sum_{t\in[1,T-1]}\left(f_{k'}(y_t, y_{t+1}, x, t) - \sum_{y_t}\sum_{y_{t+1}} P_t(y_t, y_{t+1}|x) f_{k'}(y_t, y_{t+1}, x, t)\right) - \frac{\lambda_{k'}}{\sigma^2} \quad (7)$$

Dimana :

| | |
|---|---|
| $G_k^x$ | : nilai gradien ke-k untuk fitur *node* |
| $G_{k'}^x$ | : nilai gradien ke-k untuk fitur *edge* |
| $f_k(y_t, x, t)$ | : fitur *node* ke-k |
| $f_{k'}(y_t, y_{t+1}, x, t)$ | : fitur *edge* ke-k |
| $P_t(y_t\|x)$ | : probabilitas label $y_t$ terhadap data x |
| $P_t(y_t, y_{t+1}\|x)$ | : probabilitas label $y_t$ diikuti label $y_{t+1}$ ketika data x |
| $\lambda_k$ | : parameter fitur *node* ke-k |
| $\lambda_{k'}$ | : parameter fitur *edge* ke-k |
| $\sigma$ | : Standar deviasi |

Persamaan 6 dan 7 di atas merupakan *stochastic gradient method.* Untuk menghitung $P_t(y_t|x)$ dan $P_t(y_t, y_{t+1}|x)$ digunakan prosedur *two-pass* terhadap data. Prosedur ini dikenal sebagai *forward-backward.*

1. *Forward Pass*

*Forward pass* melakukan penelusuran dari indeks t = 1 hingga t = T. Untuk menghitung seluruh probabilitas lokal diperlukan penelusuran yang berlawanan arah (*backward pass*).

Variabel *forward* dilambangkan oleh $\alpha_t[y_t]$. Ketika t =1 inisialisasi $\alpha_1[y_1] = 1/S$ untuk seluruh label $y_t$, dimana $S$ merupakan jumlah seluruh label kelas kata. Ketika $t \geq 2$ maka dapat digunakan persamaan sebagai berikut [11] :

$$\alpha_t[y_t] = k_t \sum_{x_{t-1}} \alpha_{t-1}[y_{t-1}]\phi_t(y_t, x)\psi_{t-1}(y_{t-1}, y_t, x) \quad (8)$$

Dimana :

| | |
|---|---|
| $\alpha_t[y_t]$ | : variabel *forward pass* ke-t |
| $k_t$ | : faktor penskalaan untuk memastikan $\sum_{y_t} \alpha_t[y_t] = 1$ |
| $\alpha_{t-1}[y_{t-1}]$ | : variabel *forward* pass sebelumnya |
| $\phi_t(y_t, x)$ | : nilai *node* potensial ke-t |
| $\psi_{t-1}(y_{t-1}, y_t, x)$ | : nilai *edge* potensial sebelumnya |

2. *Backward Pass*

*Backward pass* hampir sama dengan *forward pass.* Perbedaannya hanya menelusuri dari indeks t = T sampai t = 1. Variabel *Backward* dilambangkan $\beta_t[y_t]$. Ketika t = T, inisialisasi $\beta_T[y_T] = 1/S$ untuk semua label $y_T$ ketika $t < T$ maka dapat digunakan persamaan sebagai berikut [11].

$$\beta_t[y_t] = \mu_t \sum_{y_{t+1}} \beta_{t+1}[y_{t+1}]\phi_{t+1}(y_{t+1}, x)\psi_t(y_t, y_{t+1}, x) \quad (9)$$

Dimana :

| | |
|---|---|
| $\beta_t[y_t]$ | : nilai *backward pass* ke-t |
| $\mu_t$ | : faktor penskalaan untuk memastikan $\sum_{y_t} \beta_t[y_t] = 1$ |
| $\beta_{t+1}[y_{t+1}]$ | : nilai variabel *backward pass* setelahnya |
| $\phi_{t+1}(y_{t+1}, x)$ | : nilai *node* potensial setelahnya |
| $\psi_t(y_t, y_{t+1}, x)$ | : nilai *edge* potensial ke-t |

3. Probabilitas *Node* dan Probabiltas *Edge*

Untuk mendapatkan probablitas *node* yang menghitung probabilitas label kelas kata $y_t$ terhadap data masukan $x$, dapat dihitung dengan menggunakan persamaan [11] :

$$P_t(y_t|x) = w_t \alpha_t[y_t]\, \phi_t(y_t, x)\beta_t[y_t] \quad (10)$$

Dimana :

| | |
|---|---|
| $P_t(y_t\|x)$ | : probabilitas *node* ke-t |
| $w_t$ | : faktor normalisasi untuk memastikan $\sum_{y_t} P_t(y_t\|x) = 1$ |
| $\alpha_t[y_t]$ | : nilai *forward pass* variabel ke-t |
| $\phi_t(y_t, x)$ | : nilai *node* potensial ke-t |
| $\beta_t[y_t]$ | : nilai *backward pass* ke-t |

Sedangkan untuk menghitung probabilitas *edge* yang menghitung probabilitas label $y_t$ untuk data $x$ diikuti oleh label $y_{t+1}$ yang merupakan label kelas kata untuk kata setelahnya dapat dihitung menggunakan persamaan [11] :

$$P_t(y_t, y_{t+1}|x) = \gamma_t \alpha_t[y_t]\, \phi_t(y_t, x)\psi_t(y_t, y_{t+1}, x)\phi_{t+1}(y_{t+1}, x)\beta_{t+1}[y_{t+1}] \quad (11)$$

Dimana :

$P_t(y_t, y_{t+1}|x)$ : probabilitas $edge$ ke-t
$\gamma_t$ : faktor normalisasi untuk memastikan $\sum_{y_t} P_t(y_t, y_{t+1}|x) = 1$
$\alpha_t[y_t]$ : nilai *forward pass* variabel 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
$\phi_{t+1}(y_{t+1}, x)$ : nilai *node* potensial setelahnya
$\beta_{t+1}[y_{t+1}]$ : nilai *backward pass* setelahnya

### *1.3.3 Testing*

Tahap ini merupakan tahap untuk menentukan label kelas kata untuk seluruh data, bukan hanya satu *node* saja. Secara teori, untuk menemukan urutan label yang paling memungkinkan untuk diberikan terhadap data masukan. Pada langkah ini digunakan metode *Viterbi Decoding.* Ada dua langkah prosedur pada metode ini, yaitu [11] :

1. *Maximal Forward Pass*

*Maximal forward pass* digunakan untuk menemukan probabilitas paling optimal untuk data masukan dengan mempertimbangkan hubungan antara label kelas kata untuk kata yang sedang diproses dengan label kelas kata untuk kata setelahnya. Ketika t = 1, inisialisasi $\alpha_1^{max}[y_1] = 1/S$ untuk semua label $y_1$ Ketika $t \geq 2$ maka dapat digunakan persamaan sebagai berikut.

$$\alpha_t^{max}[y_t] = k_t \max_{y_{t-1}} (\alpha_{t-1}[y_{t-1}^{max}]\phi_t(y_t, x)\psi_{t-1}(y_{t-1}, y_t, x)) \quad (12)$$

Dimana :

$\alpha_t^{max}[y_t]$ : nilai maksimal untuk $\alpha_t[y_t]$
$k_t$ : faktor penskalaan untuk memastikan $\sum_{y_t} \alpha_t^{max}[y_t] = 1$
$\alpha_{t-1}[y_{t-1}^{max}]$ : nilai maksimal untuk label kelas kata untuk kata sebelumnya
$\phi_t(y_t, x)$ : nilai *node* potensial ke-t
$\psi_{t-1}(y_{t-1}, y_t, x)$ : nilai *edge* potensial sebelumnya

2. *Backtracking Pass*

Proses *backtracking* digunakan untuk menemukan label paling optimal. Adapun persamaannya adalah sebagai berikut.

$$y_t^* = \arg\max_{y_t}\left(\alpha_t^{max}[y_t]\phi_t(y_t, x)\right) \quad (13)$$

Dimana :

$\alpha_t^{max}[y_t]$ : nilai maksimal untuk $\alpha_t[y_t]$
$\phi_t(y_t, x)$ : nilai node potensial ke-t

## 2. ISI PENELITIAN

### 2.1 Tagset

*Tagset* atau *list* kelas kata memiliki banyak kelas kata terutama untuk bahasa Inggris, seperti : 45 untuk *Penn Treebank*, 87 untuk *Brown Corpus*, dan 146 untuk *C7 tagset* [2]. Sementara untuk bahasa Indonesia sendiri, *tagset* yang digunakan lebih sedikit karena tidak semua *tagset* yang ada di bahasa Inggris dapat diimplementasikan ke bahasa Indonesia. Ada beberapa *tagset* pada Bahasa Indonesia yaitu, 25 *tagset* [12], 19 *tagset* [13] serta 23 *tagset* [2]. *Tagset* yang digunakan pada penelitian ini adalah 23 *tagset* [2], yaitu :

Tabel 2 *Tagset*

| No. | *Tag* | Deskripsi |
|---|---|---|
| 1 | CC | Konjungtor koordinatif |
| 2 | CD | Numeralia kardinal |
| 3 | OD | Numeralia ordinal. |
| 4 | DT | Artikel atau disebut juga artikula. |
| 5 | FW | Kata bahasa asing. |
| 6 | IN | Preposisi atau kata depan. |
| 7 | JJ | Adjektiva |
| 8 | MD | Verba modal dan verba bantu |
| 9 | NEG | Kata ingkar |
| 10 | NN | Nomina, atau disebut juga kata benda. |
| 11 | NNP | *Proper noun.* |
| 12 | NND | Penggolong atau nomina ukuran. |
| 13 | PR | Deemonstrativa atau pronomina penunjuk |
| 14 | PRP | Pronomina persona. |
| 15 | RB | Adverbia atau disebut juga kata keterangan |
| 16 | RP | Partikel |
| 17 | SC | Konjungtor subordinatif |
| 18 | SYM | Simbol. |
| 19 | UH | Interjeksi |
| 20 | VB | Verba disebut juga kata kerja |
| 21 | WH | Pronomina penanya |
| 22 | X | Kata atau bagian dari kalimat yang tidak diketahui atau belum diketahui secara pasti kategorinya. |
| 23 | Z | Tanda baca. |

### 2.2 Data Masukan

Berikut ini merupakan contoh data masukan yang digunakan yang berasal dari korpus [2] :

Curah/NN
hujan/NN
yang/SC
tinggi/JJ
menyebabkan/VB
banyak/CD
genangan/NN
,/Z
tempat/NN
ideal/JJ
bagi/IN
nyamuk/NN
aedes/NNP
berkembang biak/VB
./Z

Pemkot/NN
telah/MD
menempatkan/VB
ikan/NN
di/IN
berbagai/CD
genangan/NN
air/NN
untuk/SC
memangsa/VB
telur/NN
nyamuk/NN
./Z

Gambar 1 Contoh Data Masukan

### 2.3 Praproses

Sebelum masuk ke metode utama dalam proses *POS tagging*, dilakukan praproses terlebih dahulu. Adapun praproses yang dilakukan adalah sebagai berikut.

#### *2.3.1 Case Folding*

Tahap *case folding* merupakan pengubahan karakter huruf pada teks menjadi huruf kecil semua. Hal ini dilakukan untuk mempermudah dalam pencarian kata di data *training*.

#### 2.3.2 Tokenisasi Kalimat

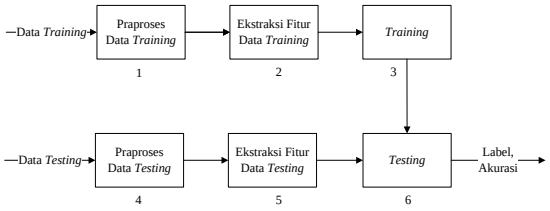
Tokenisasi kalimat merupakan pemisahan teks kedalam bentuk kalimat-kalimat dengan baris kosong sebagai pemisah antar kalimat.

#### 2.3.3 Tokenisasi Kata

Tokenisasi (*tokenization*) merupakan pemisahan kata-kata dari kalimat, pemisahan ini dilakukan dengan *delimiter* spasi, tab, dan *new line*. Sementara titik (.), koma(,) dan tanda baca lainnya masuk dalam kategori kata.

### 2.3 Analisis Proses

Berikut ini adalah gambaran sistem untuk proses pelabelan kelas kata yang akan dibangun, yaitu :




Gambar 2 Proses POS *Tagging*

## 3 PENGUJIAN

### 3.1 Pengujian Nilai Parameter

Pada tahap ini akan dilakukan pengujian pengaruh nilai parameter terhadap akurasi yang dihasilkan. Adapun nilai parameter yang diuji adalah inisialisasi nilai awal parameter ($\lambda$) , nilai standar deviasi ($\sigma$) serta nilai *learning rate* ($\omega$). Untuk inisialisasi nilai awal nilai yang akan diuji adalah 0 dan 1, nilai standar deviasi rentang nilainya adalah antara 2 sampai 8, sedangkan nilai *learninng rate* antara 0,01 sampai 0,000001. Data *Training* yang digunakan berisi 200 kalimat yang terdiri dari 3910 kata yang berasal dari korpus, sedangkan data *testing* yang digunakan berisi 50 kalimat yang memiliki 908 kata. Berikut ini merupakan hasil pengujian nilai parameter.

Tabel 3 Hasil Pengujian Nilai Parameter

| Nilai Awal Parameter Fitur | Learning rate | Standar Deviasi | Akurasi |
|---|---|---|---|
| 0 | 0,01 | 2 | 0% |
| | | 4 | 0% |
| | | 8 | 0% |
| | 0,0001 | 2 | 60,79% |
| | | 4 | 60,79% |
| | | 8 | 60,79% |
| | 0,000001 | 2 | 60,79% |
| | | 4 | 60,79% |
| | | 8 | 60,79% |
| 1 | 0,01 | 2 | 0% |
| | | 4 | 0% |
| | | 8 | 0% |
| | 0,0001 | 2 | 77,75 % |
| | | 4 | 77,75 % |
| | | 8 | 77,75 % |

### 3.2 Pengujian Akurasi

Berikut ini merupakan hasil pengujian akurasi yang dilakukan terhadap 4 data *testing* yang masing-masing terdiri dari 50 kalimat dari hasil *training* 200 kalimat yang memiliki 3910 kata. Adapun nilai parameter yang digunakan dalam *training* merupakan nilai yang menghasilkan akurasi paling baik dari proses pengujian nilai parameter. Untuk nilai inisialisasi awal parameter fitur adalah 1, nilai standar deviasi adalah 2 serta nilai *learning rate* adalah 0,000001. Adapun untuk penghitungan akurasi untuk tiap data *testing* adalah sebagai berikut.

$$\text{Akurasi} = \frac{\text{Jumlah prediksi label kelas kata yang benar}}{\text{Jumlah kata pada data testing}} \text{ x } 100\%$$

Berikut ini merupakan perhitungan akurasi untuk Data *Testing* 1.

$$\text{Akurasi} = \frac{\text{Jumlah prediksi label kelas kata yang benar}}{\text{Jumlah kata pada data testing}} \text{x } 100\%$$
$$= \frac{706}{908}\ x\ 100\ \%$$
$$= 77{,}75\ \%$$

Adapun untuk hasil pengujian untuk seluruh data *testing* dapat dilihat pada tabel berikut.

Tabel 4 Hasil Pengujian Akurasi

| Data *Testing* | Jumlah Kata | Jumlah Prediksi Label Kelas Kata yang Benar | Akurasi |
|---|---|---|---|
| 1 | 908 | 978 | 77,75% |
| 2 | 949 | 732 | 77,13 % |
| 3 | 1009 | 793 | 78,59% |
| 4 | 2902 | 1467 | 79,31% |
| **Rata-rata** | | | 78,20% |

## 5. KESIMPULAN & SARAN

Berdasarkan pada hasil pengujian yang dilakukan pada penelitian tugas akhir ini, rata-rata akurasi yang dihasilkan oleh POS *Tagger* untuk Bahasa Indonesia dengan menggunakan CRF adalah sebesar 78,20 %.
Adapun saran dalam pengembangan POS *Tagger untuk* Bahasa Indonesia dengan menggunakan CRF ini adalah sebagai berikut.

1. Penambahan fitur *node* yang dapat meningkatkan probabilitas suatu kata diberikan label tertentu.
2. Pengimplementasian metode *training* lain untuk ,mengetahui metode *training* yang lebih akurat untuk digunakan pada POS *tagger* Bahasa Indonesia dengan menggunakan CRF.
3. Optimasi program sehingga bisa memproses lebih banyak data *training*.

## 6. DAFTAR PUSTAKA


[1] Jurafsky, D, Martin, J.H. Speech and Language Processing : An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. New Jersey : Prentice-Hall Inc. 2000.

[2] Dinakaramani, A, Et. al. *Designing an Indonesian Part of Speech Tagset and Manually Tagged Indonesian Corpus*. 2014.

[3] Pisceldo, F, Adriani, M, Manurung, R. *Probabilistic Part of Speech Tagging for Bahasa Indonesia*. 2009.

[4] Wicaksono, A.F., Purwanti, A. HMM Based POS Tagger for Bahasa Indonesia, Fourth International MALINDO Workshop.

[5] Chandrawati, T. Pengembangan Part of Speech Tagger untuk Bahasa Indonesia Berdasarkan Metode Conditional Random Fields dan Transformation Based Learning. Universitas Indonesia. 2008.

[6] Rashel, F, Luthfi, A, Dinakaramani, A, Manurung, R. Building an Indonesian Rule-Based Part-of-Speech Tagger. 2014.

[7] Manurung, R. Tutorial: Pengenalan terhadap POS Tagging dan probabilistic Parsing. Workshop Nasional INACL, Jakarta. 2016.

[8] Lafferty, J, McCallum, A, Pereira, F. Conditional Random Field: Probabilistic Model for Segmenting and Labeling Sequence Data. 2001.

[9] Wallach, M. Hanna. Conditional Random Fields : An Introduction. 2004.

[10] Klinger, R, Tomanek, K. Classical Probabilistic Model and Conditional Random Field. 2007.

[11] Truyen, T.T, Phung, D. A Practitioner Guide to Conditional Random Field for Sequential Labeling. 2008.

[12] M. Adriani, R. Manurung, and F. Pisceldo, "Statistical Based Part Of Speech Tagger for Bahasa Indonesia", in Proceedings of the 3rd International MALINDO Workshop, Co-located Event ACL-IJCNLP 2009. Singapore, August 1,2009.

[13] S. D. Larasati, V. Kuboň, and D. Zeman, "Indonesian Morphology Tool (MorphInd): Towards an Indonesian Corpus", in Proceedings of the Workshop on Systems and Frameworks for Computational Morphology (SFCM 2011). August 2011. Zurich, Switzerland

[14] Kridalaksana, H. Kelas Kata Dalam Bahasa Indonesia Edisi 2. Jakarta : Gramedia. 2008.