Jurnal Linguistik Komputasional
JLK
Vol 3, No 1
Maret 2020

Pencarian Kesamaan Redaksional pada Terjemahan Al-Quran Bahasa Indonesia Menggunakan Metode Rule-based Chunking

Alfredo Primadita*†, Moch. Arif Bijaksana *§, Eko Darwiyanto *¶
*Jurusan Teknik Informatika, Fakultas Informatika, Universitas Telkom
Jl. Telekomunikasi Terusan Buah Batu, Sukapura, Kec. Dayeuhkolot, Bandung, Jawa Barat, Indonesia
*alfredoif@students.telkomuniversity.ac.id
¶ekodarwiyanto@telkomuniversity.ac.id
§arifbijaksana@telkomuniversity.ac.id

***Abstrak*—Al-Quran merupakan kitab suci yang menjadi panutan dan rujukan seluruh umat Muslim di dunia. Al-Quran terdiri dari 114 surat, 6236 ayat, dan 77845 kata. Dikarenakan Al-Quran memiliki banyak informasi di dalamnya, salah satu upaya untuk mempermudah pembaca dalam mempelajari dan menyaring informasi dari Al-Quran adalah dengan membuat parafrasa yang singkat dengan mempergunakan sistem *Rule-based Chunking*. *Rule-based Chunking* dapat mempersingkat suatu kalimat dengan mengelompokan kata-kata yang dianggap penting berdasarkan aturan tata bahasa. Sistem memiliki input berupa ayat-ayat Al-Quran terjemahan bahasa Indonesia yang sudah dicari *Longest Common Substring* dan *Longest Common Subsequence* dengan output berupa himpunan parafrasa. Dari 2341 teks berulang yang didapatkan pada penelitian sebelumnya, telah diperoleh 1261 *chunk*. Setelah melakukan evaluasi pada sistem, akurasi yang didapatkan adalah 91%, dengan *precision* 63,5% dan *recall* 74,4%.**

***Kata Kunci*—Al-Quran, shallow parsing, kesamaan redaksional, Rule-based Chunking, parafrasa.**

I. Pendahuluan

Al-Quran merupakan kitab suci yang menjadi panutan dan rujukan seluruh umat Muslim di dunia. Dalam era modern sekarang ini, Al-Quran dapat dijumpai dalam berbagai macam media digital. Al-Quran terdiri dari 114 Surat, 6236 ayat, dan 77845 kata. Dikarenakan isi Al-Quran yang banyak, pembaca relatif susah untuk mempelajari Al-Quran secara menyeluruh dan memiliki kemungkinan untuk melewatkan informasi penting yang ada di Al-Quran. Salah satu upaya untuk mempermudah pembaca dalam mempelajari dan menyaring informasi dari Al-Quran adalah dengan melakukan pencarian kesamaan redaksional.

Proses pencarian kesamaan redaksi dimulai dengan melakukan identifikasi terhadap terjemahan ayat-ayat Al-Quran yang mempunyai kemiripan. Selanjutnya redaksi terjemahan ayat-ayat Al-Quran yang mempunyai kemiripan itu dipilah lagi untuk menentukan jenis kemiripan yang terkandung di dalamnya: apakah kemiripan lafal (redaksi) atau makna [1]. Untuk mencari kesamaan redaksi pada penelitian ini, telah dilakukan pemrosesan *Longest Common Substring* dan *Longest Common Subsequence*. Akan tetapi hasil dari pemrosesan tersebut belum sesuai dengan ekspektasi yang diharapkan, pemotongan kalimat masih tidak sempurna dan sering kali artinya tidak sesuai dengan kalimat sebenarnya. Sehingga dibutuhkan proses lebih lanjut yaitu melalui *shallow parsing*. Dengan *shallow parsing*, pemotongan kalimat dilakukan dalam bentuk himpunan frasa sehingga kalimat dipotong dengan tepat dan tidak mengubah arti sebenarnya.

Shallow parsing biasa disebut juga dengan *chunking*. Metode *chunking* memiliki dua pendekatan yaitu pendekatan *machine learning* dan pendekatan *rule-based* [2]. Dalam penelitian ini, penulis menggunakan pendekatan *rule-based chunking*. *Rule-based chunking* dapat menyaring suatu kalimat dengan mengelompokan kata-kata yang dianggap penting menggunakan aturan-aturan berdasarkan aturan tata bahasa sehingga kalimat-kalimat tersebut dapat dimengerti dengan mudah.

Proses penyaringan pada penelitian ini menggunakan *Natural Language Toolkit* (NLTK) dan dilakukan melalui beberapa tahap, yaitu membagi kata pada masing-masing kalimat, memprediksi *Part of Speech* (POS) dari kata-kata tersebut, mencari dan mengelompokan kata sesuai dengan aturan *chunker*. Sistem memiliki input *data set* berupa terjemahan ayat-ayat Al-Quran yang sudah dicari *Longest Common Substring* [3] dan *Longest Common Subsequence* [4] oleh penelitian sebelumnya dengan output berupa himpunan frasa.

II. Tinjauan Pustaka

Pada penelitian S.K. Saritha dan R.K. Pateriya [5] menyebutkan bahwa analisis sintaksis atau *parsing* dibagi menjadi dua kategori yang berbeda yaitu: *full parsing* dan *shallow parsing* (*partial parsing* atau *chunk parsing*). *Full parsing* membangun *parse tree* yang lengkap untuk sebuah kalimat dimana menyebabkan perolehan akurasi yang rendah. *Full Parsing* menjadi sangat lambat karena prosesor-intensif dan memori-intensif. Sedangkan *Shallow*

Submitted 26-02-2020; Revised 17-03-2020; Accepted 26-03-2020

parsing membagi kalimat-kalimat ke dalam urutan *chunk* yang merupakan bagian teks yang tidak tumpang tindih. Dengan menggunakan teknik *rule-based shallow parsing* pada identifikasi kalimat komparatif dari konten yang dibuat pengguna, memperoleh *precision* 84.1% dan *recall* 72.2% untuk *nonequal gradable*, 82.2% dan 87.6% untuk *superlatives*.

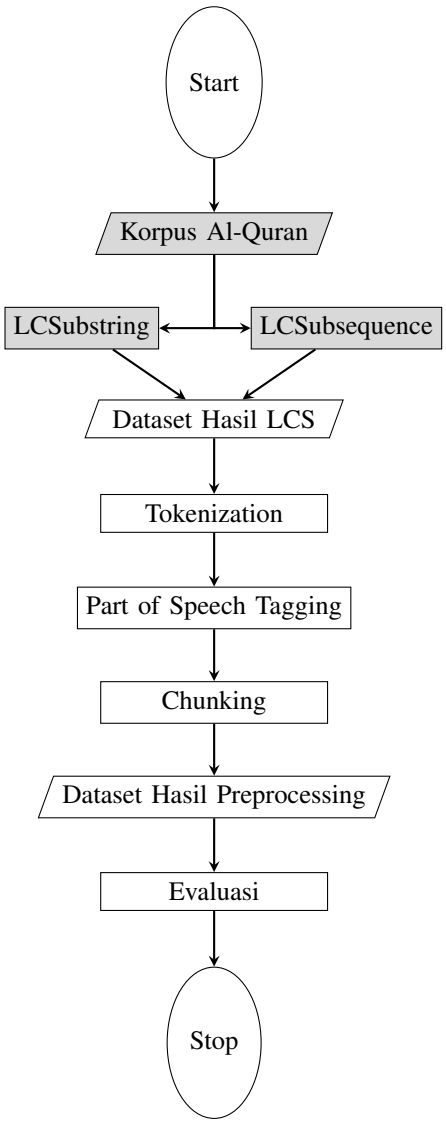
Penelitian Nabil Khoufi, Chafik Aloulou dan Lamia Hadrich Belguith berasumsi bahwa melakukan *chunking* dengan pendekatan *learning* lebih bermanfaat daripada pendekatan *rule-based* dalam pembuatan tata bahasa Arab, karena bahasa Arab memiliki struktur tata bahasa yang kompleks [6]. Penelitian tersebut menggunakan metode *supervised learning* untuk men-*chunking* teks bahasa Arab, yaitu *Conditional Random Field*. Keunggulan dari CRF dibandingkan dengan model klasifikasi konvensional adalah mempertimbangkan ketergantungan antara anotasi yang saling berhubungan dalam grafik. Namun demikian, pertimbangan ini memiliki kelemahan yang harus dibayar yaitu: fase pembelajaran menggunakan CRF bisa memakan waktu lama. Dengan test set berisi 2524 kalimat, penelitian ini menghasilkan akurasi token 96.54%.

Penelitian [7] mengusulkan metodologi untuk menerapkan *shallow parsing* yang tangguh serta memiliki cakupan yang luas, dengan menggunakan kombinasi antara *POS Tagging* berbasis *machine learning* (Maximum Entropy) dan *rule-based chunker* tanpa membutuhkan korpus beranotasi yang besar. Pendekatan *rule-based* merupakan salah satu pendekatan dari *chunking*. Pendekatan *rule-based* adalah pendekatan yang membutuhkan usaha yang intensif, membosankan, lambat, rawan kesalahan. Pendekatan *rule-based* diambil dengan keyakinan kuat bahwa itu akan menghasilkan *dataset* yang dapat digunakan untuk penelitian masa depan tanpa menyiapkan korpus *chunk* beranotasi secara manual [7]. *POS Tagger* berbasis Maximum Entropy yang telah dibuat pada penelitian ini menghasilkan akurasi 81.72%, akurasi tersebut dapat ditingkatkan dengan memperbesar ukuran *training* korpus. Untuk *rule-based chunker* sendiri menghasilkan 78.3%. Sedangkan untuk kombinasi dari *POS Tagger* berbasis Maximum Entropy dan *rule-based chunker* menghasilkan 66.6%, karena kesalahan dari *POS Tagger* digunakan pada *rule-based chunker*, menyebabkan turunnya performa akhir dari *shallow parser.*

Pada penelitian C. Grover dan R. Tobin menyebutkan bahwa permasalah *reusability* dan *portability* adalah masalah penting yang menjadi landasan dalam menentukan pendekatan apa yang akan digunakan. Pendekatan *machine learning chunker* akan membutuhkan materi *training* beranotasi yang memerlukan proses lebih panjang, tetapi sistem *machine learning* dapat mengesampingkan keputusan *POS tagging* dimanapun fitur yang lain menyarankannya. Sedangkan pendekatan sistem *rule-based chunker* akan membutuhkan modifikasi yang relatif kecil untuk dapat digunakan kembali pada permasalahan yang berbeda, akan tetapi sistem *rule-based* bergantung pada *POS tagging* yang baik dan akan berkinerja buruk apabila penandaan data dilakukan dengan buruk [2].

Berdasarkan penelitian-penelitian tersebut penulis mengusulkan untuk menggunakan *POS tagging* berbasis *machine learning* yaitu *Conditional Random Field* (CRF) dan dikombinasi dengan *rule-based chunker*, karena mempertimbangkan tingkat *reusability* sehingga sistem dan data yang diperoleh dapat dengan mudah dimodifikasi dan digunakan kembali pada penelitian selanjutnya.

III. Metode Penelitian



Gambar. 1. *Flowchart* gambaran umum sistem. Simbol (Input, LCSubstring, dan LCSubsequence) yang berwarna abu-abu merupakan proses dari penelitian sebelumnya sehingga tidak dibahas pada penelitian ini.

A. Data Set

Data set yang digunakan pada penelitian ini ialah ayat-ayat Al-Quran yang telah diproses dengan *longest common*

substring [3] dan *longest common subsequence* [4] dari penelitian sebelumnya yang berisi kata-kata yang memiliki kesamaan redaksi pada ayat-ayat Al-Quran, beberapa contoh dapat dilihat pada tabel I dan II.

TABEL I
HASIL TEKS BERULANG DARI PROSES LONGEST COMMON SUBSTRING

Results	QS-1	QS-2	Total
maha pemurah lagi maha penyayang	(1:1)	(1:1), (1:3), (2:163)	3
lagi maha penyayang	(1:1)	(1:1), (1:3), (2:37), (2:54), (2:128), (2:143), (2:160), (2:163), (2:173), (2:182), (2:192), (2:218), (2:226), (3:31), (3:89), (3:129), (4:16), (4:23), (4:25), (4:64), (2:199), (4:96), (4:100), (4:106), (4:110), (4:129), (4:152), (5:3), (5:34), (5:39), (5:74), (5:98)	32
menyebut nama allah	(1:1)	(2:114)	2
tuhan semesta alam	(1:2)	(2:131)	2
jalan yang lurus	(1:6)	(1:6), (2:108), (2:142), (2:213), (3:51), (3:101), (4:68), (4:137), (4:175), (5:12), (5:16), (5:60), (5:77)	13
...	...	...	...

TABEL II
HASIL TEKS BERULANG DARI PROSES LONGEST COMMON SUBSEQUENCE

Results	QS-1	QS-2	Total
maha pemurah maha penyayang	(1:1)	(1:3), (2:163)	3
allah maha maha penyayang	(1:1)	(2:143), (2:173), (2:182), (2:192), (2:199), (2:218), (2:226), (2:37), (2:54), (3:129), (3:31), (3:89), (4:100), (4:106), (4:110), (4:129), (4:152), (4:16), (4:23), (4:25), (4:64), (4:96), (5:3), (5:34), (5:39),(5:74), (5:98)	28
menyebut allah maha maha	(1:1)	(2:235)	2
maha maha penyayang	(1:3)	(2:128), (2:143), (2:160), (2:173), (2:182), (2:192), (2:199), (2:218), (2:226), (2:37), (2:54), (3:129), (3:31), (3:89), (4:100), (4:106), (4:110), (4:129), (4:152), (4:16), (4:23), (4:25), (4:64), (4:96), (5:3), (5:34), (5:39), (5:74), (5:98)	30
alif laam miim	(2:1)	(3:1)	2
...	...	...	...

B. Preprocessing

Sebelum *data set* di-*chunking*, beberapa tahap *preprocessing* dilakukan, seperti tokenisasi dan pelabelan *part-of-speech*. Label tersebut digunakan pada proses *chunking* untuk mencari susunan kata yang sesuai dengan aturan *chunker* yang dibuat berdasarkan aturan tata bahasa.

1) Tokenization: Pada tahap tokenisasi, data hasil penelitian sebelumnya dijabarkan dan diproses pada setiap baris. Input setiap baris dipecah menjadi per kata. Berikut ini contoh tokenisasi yang diterapkan.
Input hasil LCSubstring (2:62 dan 5:69):
sesungguhnya orang-orang mukmin orang-orang yahudi
Output:
”sesungguhnya”, ”orang-orang”, ”mukmin”, ”orang-orang”, ”yahudi”

2) Part of Speech Tagging: POS Tagging dilakukan berdasarkan aturan tata bahasa dan dianotasikan secara manual oleh penulis. Tagset yang digunakan pada penelitian ini merupakan hasil dari penelitian Arawinda Dinakaramani, Fam Rashel, Andry Luthfi, dan Ruli Manurung [8]. Korpus Tagset bahasa Indonesia telah dimodifikasi untuk menyesuaikan data yang digunakan pada penelitian ini.

TABEL III
HASIL SETELAH MELALUI PROSES POS TAGGING

Hasil dari proses pelabelan Part-of-Speech
[(’maha’, ’RB’), (’pemurah’, ’JJ’), (’lagi’, ’RB’), (’maha’, ’RB’), (’penyayang’, ’JJ’)]
[(’lagi’, ’RB’), (’maha’, ’RB’), (’penyayang’, ’JJ’)]
[(’menyebut’, ’VB’), (’nama’, ’NN’), (’allah’, ’NNP’)]
[(’tuhan’, ’NN’), (’semesta’, ’NN’), (’alam’, ’NN’)]
[(’jalan’, ’NN’), (’yang’, ’SC’), (’lurus’, ’JJ’)]
[(’maha’, ’RB’), (’pemurah’, ’JJ’), (’maha’, ’RB’), (’penyayang’, ’JJ’)]
[(’allah’, ’NNP’), (’maha’, ’RB’), (’maha’, ’RB’), (’penyayang’, ’JJ’)]
[(’menyebut’, ’VB’), (’allah’, ’NNP’), (’maha’, ’RB’), (’maha’, ’RB’)]
[(’maha’, ’RB’), (’maha’, ’RB’), (’penyayang’, ’JJ’)]
[(’alif’, ’FW’), (’laam’, ’FW’), (’miim’, ’FW)]
...

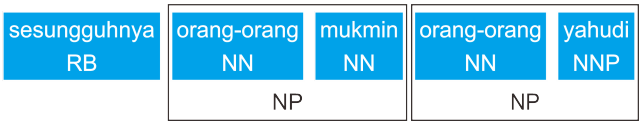
Pada tahap POS Tagging, setiap kata di setiap baris diberi *tag* sehingga akan menjadi ”sesungguhnya/RB orang-orang/NN mukmin/NN orang-orang/NN yahudi/NNP”, seperti ilustrasi pada gambar 2 dan beberapa contoh pada tabel III.

Gambar. 2. Kata-kata pada setiap baris diberi *tag*

C. Chunking / Shallow Parsing

Setelah setiap baris ditokenisasi dan setiap kata diberi *tag* pada tahap *preprocessing*, sistem akan mencari dan mengelompokan kata yang sesuai dengan aturan *chunker*. Aturan-aturan tersebut dibuat sesuai dengan pola tata bahasa Indonesia dan menghindari pola yang ambigu, sehingga menjadi; seperti ilustrasi pada gambar 3.
Contoh Aturan Chunker;

NP: {<NN><CC>?<NN>}
{<NN>?<NNP>+}
Output:
(S sesungguhnya/RB
(NP orang-orang/NN mukmin/NN)
(NP orang-orang/NN yahudi/NNP))



Gambar. 3. Ilustrasi kalimat setelah di *chunking*

IV. Hasil dan Pembahasan

A. Test Set

Test set merupakan himpunan frasa dari data terjemahan Al-Quran yang telah dibuat dan diperiksa kebenarannya secara manual.

B. Hasil Pengujian

Pengujian dilakukan dengan membandingkan hasil atau *output* dengan *test set* yang berupa *gold standard* yang telah dibuat secara manual seperti pada Tabel IV atau Tabel V, sehingga dapat diketahui berapa nilai *accuracy, precision, recall* dan *F-measure*.

TABEL IV
CONTOH PERHITUNGAN *True Positive* (TP), *False Positive* (FP), *False Negative* (FN), DAN *Correct Tokens* (CT) DENGAN MEMBANDINGKAN HASIL DAN GOLD STANDARD.

Input	Output	Gold standard	CT	TP	FP	FN
menyebut nama allah	*(S (NP menye-but/VB nama/NN) (NP allah/NNP))*	*(S menye-but/VB nama/NN (NP allah/NNP))*	3	1	1	0
alif laam miim	*(S alif/FW laam/FW miim/FW)*	*(S (NP alif/FW laam/FW miim/FW))*	3	0	0	1

Dengan membandingkan *output* dan *gold standard*, dapat diketahui nilai CT, TP, FP, FN. Nilai *Correct Tokens* (CT) didapatkan dengan menjumlah setiap label token pada *output* yang sesuai dengan *gold standard*, sebagai contoh data pertama pada Tabel IV.
Output: *menyebut/VB nama/NN allah/NNP*
Gold standard: *menyebut/VB nama/NN allah/NNP*

Setiap label token pada *output* sesuai dengan label token pada *gold standard* dengan jumlah seluruh token adalah tiga, sehingga nilai CT adalah tiga. Nilai *True Positive* (TP) diperoleh dengan menjumlah *chunk* yang diprediksi dengan benar atau sesuai dengan *gold standard*, sebaliknya, *False Positive* (FP) diperoleh dengan menjumlah *chunk* yang telah diprediksi akan tetapi salah atau tidak sesuai dengan *gold standard*, sebagai contoh data pertama pada Tabel IV.
Output: *(NP VB NN) (NP NNP)*
Gold standard: *VB NN (NP NNP)*

Output memiliki dua *chunk*, sedangkan pada *gold standard* hanya satu *chunk*, sehingga masing-masing nilai TP dan FP adalah satu dan *chunk* yang benar hanya *(NP NNP)*. Sedangkan untuk nilai *False Negative*

TABEL V
BEBERAPA CONTOH DATA YANG DIEVALUASI, DENGAN MEMBANDINGKAN HASIL DAN GOLD STANDARD.

Input	Output	Gold standard	CT	TP	FP	FN
maha pemurah lagi maha penyayang	*(S (NP maha/RB pemu-rah/JJ) lagi/RB (NP maha/RB penyayang/JJ))*	*(S (NP maha/RB pemu-rah/JJ) lagi/RB (NP maha/RB penyayang/JJ))*	5	2	0	0
lagi maha penyayang	*(S lagi/RB (NP maha/RB penyayang/JJ))*	*(S lagi/RB (NP maha/RB penyayang/JJ))*	3	1	0	0
menyebut nama allah	*(S (NP menye-but/VB nama/NN) (NP allah/NNP))*	*(S menye-but/VB nama/NN (NP allah/NNP))*	3	1	1	0
tuhan semesta alam	*(S tuhan/NN (NP semesta/NN alam/NN))*	*(S (NP tuhan/NN semesta/NN alam/NN))*	3	0	1	0
jalan yang lurus	*(S (NP jalan/NN yang/SC lurus/JJ))*	*(S (NP jalan/NN yang/SC lurus/JJ))*	3	1	0	0
maha pemurah maha penyayang	*(S (NP maha/RB pemu-rah/JJ)(NP maha/RB penyayang/JJ))*	*(S (NP maha/RB pemu-rah/JJ)(NP maha/RB penyayang/JJ))*	4	2	0	0
allah maha maha penyayang	*(S (NP allah/NNP) maha/RB (NP maha/RB penyayang/JJ))*	*(S (NP allah/NNP) maha/RB (NP maha/RB penyayang/JJ))*	4	2	0	0
menyebut allah maha maha	*(S menye-but/VB (NP allah/NNP) maha/RB maha/RB)*	*(S menye-but/VB (NP allah/NNP) maha/RB maha/RB)*	4	1	0	0
maha maha penyayang	*(S maha/RB (NP maha/RB penyayang/JJ))*	*(S maha/RB (NP maha/RB penyayang/JJ))*	3	1	0	0
alif laam miim	*(S alif/FW laam/FW miim/FW)*	*(S (NP alif/FW laam/FW miim/FW))*	3	0	0	1
…	…	…	…	…	…	…

(FN) didapatkan dengan menjumlah *chunk* yang tidak terprediksi oleh sistem, sebagai contoh data kedua pada Tabel IV.
Output: *alif/FW laam/FW miim/FW*
Gold standard: *(NP alif/FW laam/FW miim/FW)*

Pada *output*, sistem tidak memprediksi adanya *chunk*, tetapi seharusnya terdapat satu chunk yaitu (NP FW FW FW), sehingga didapatkan nilai satu pada FN. Setelah seluruh test set diketahui nilai *Correct Tokens* (CT), TP, FP, dan FN, nilai-nilai tersebut dijumlah untuk digunakan pada perhitungan *precision, recall, dan* F_1*-score*.

C. Analisis Hasil Pengujian

Pengujian dilakukan dengan membandingkan hasil atau *output system* dengan *test set gold standard* yang telah dibuat secara manual seperti pada tabel III, sehingga dapat diketahui berapa nilai *accuracy, precision, recall* dan *F-measure*. Nilai akurasi didapat dengan menghitung jumlah token yang benar dibagi dengan jumlah keseluruhan token. Token adalah kata di setiap baris yang ada pada *data set*.

Himpunan parafrasa telah berhasil dibangun dengan cukup baik, akan tetapi memiliki banyak ruang untuk dapat ditingkatkan kembali. Seperti pada aspek data training untuk POS Tagging, serta kompleksitas aturan-aturan chunker.

Tabel VI merupakan hasil pengujian dari *data set* berjumlah 2341 teks berulang yang telah diproses menggunakan algoritma *rule-based chunking* dan dibandingkan dengan *test set*. Untuk nilai *precision, recall,* serta F_1*-score* relatif rendah karena aturan-aturan *chunker* dan juga training data untuk pelabelan *part-of-speech* masih perlu diperinci dan diperbanyak lagi. Untuk aturan *chunker* yang paling sering dipakai adalah {<PRP>?<NEG>?<VB><NN>+}, karena banyak ditemukan *chunk* yang memiliki pelablean <VB> diikuti dengan <NN>dengan jumlah kurang lebih 30% dari total data.

TABEL VI
NILAI HASIL PENGUJIAN MASING-MASING PARAMETER

	Value
Precision	63,5%
Recall	74,4%
F_1-score	68,5%

V. KESIMPULAN

1) Sistem *rule-based chunking* dapat melakukan pengelompokan frasa dengan cukup efisien untuk bahasa yang memiliki struktur tata bahasa relatif simpel.
2) *Training data* pada proses *part-of-speech tagging* membutuhkan waktu lebih lama seiring dengan banyaknya *training data*.
3) Pelabelan *part-of-speech* akan mempengaruhi hasil dari *rule-based chunking*, karena proses dari *rule-based chunking* yang menggunakan label tersebut untuk mengelompokan kata-kata sesuai aturan *chunker*.
4) Kompleksitas aturan chunker dapat mempengaruhi hasil keluaran sistem.

REFERENSI

[1] N. Baidan, *Metode penafsiran al-Qur'an: kajian kritis terhadap ayat-ayat yang beredaksi mirip*. Pustaka pelajar, 2011.
[2] C. Grover and R. Tobin, "Rule-based chunking and reusability." in *LREC*, 2006, pp. 873–878.
[3] D. Oktaviani, M. A. Bijaksana, and I. Asror, "Building a database of recurring text in the Quran and its translation," *Procedia Computer Science*, vol. 157, pp. 125–133, 2019.
[4] M. A. Rasyid, M. A. Bijaksana, and I. Asror, "Building of similar common subsequence text corpus of quran," *IND. Journal On Computing*, 2019.
[5] S. Saritha and R. Pateriya, "Rule-based shallow parsing to identify comparative sentences from text documents," in *Emerging Research in Computing, Information, Communication and Applications*. Springer, 2016, pp. 355–365.
[6] N. Khoufi, C. Aloulou, and L. H. Belguith, "Chunking arabic texts using conditional random fields," in *2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA)*. IEEE, 2014, pp. 428–432.
[7] I. Ariaratnam, A. Weerasinghe, and C. Liyanage, "A shallow parser for tamil," in *2014 14th International Conference on Advances in ICT for Emerging Regions (ICTer)*. IEEE, 2014, pp. 197–203.
[8] A. Dinakaramani, F. Rashel, A. Luthfi, and R. Manurung, "Designing an Indonesian part of speech tagset and manually tagged indonesian corpus," in *2014 International Conference on Asian Language Processing (IALP)*. IEEE, 2014, pp. 66–69.