

BAB 3 ARSITEKTUR DAN PERANCANGAN SISTEM

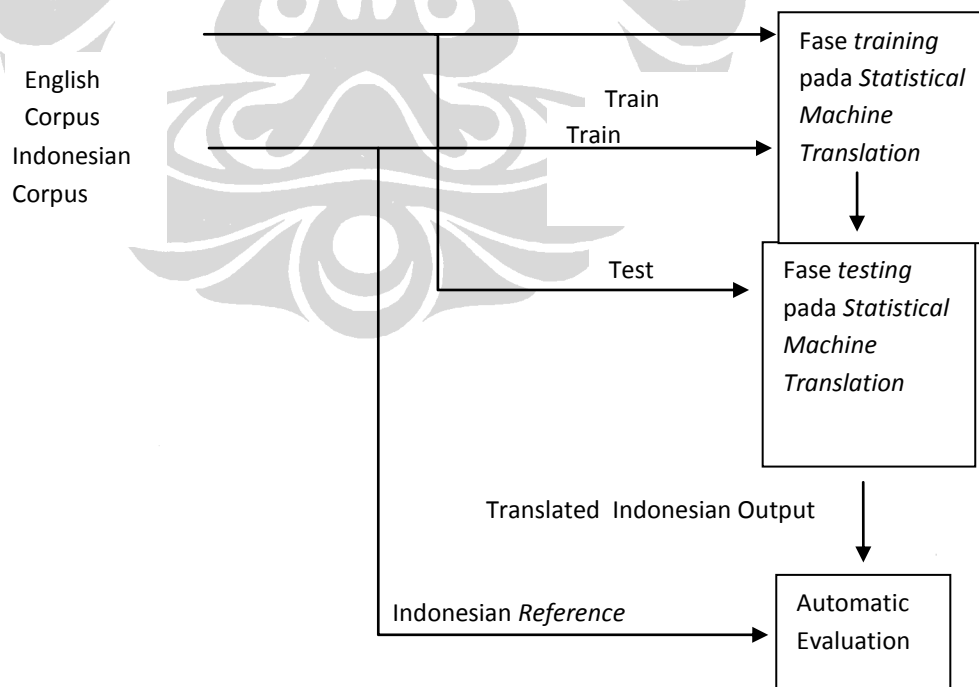
Bab ini menjelaskan tentang arsitektur dan perancangan sistem pada saat pengerjaan tugas akhir. Pembahasan dimulai dari arsitektur sistem (subbab 3.1), korpus yang digunakan (subbab 3.2), *K-fold cross validation* (subbab 3.3), rancangan *word reordering* (subbab 3.4), dan rancangan *phrase reordering* (subbab 3.5). Subbab pada bab ini penulis susun dalam urutan sistematis, sesuai dengan alur yang penulis lalui selama masa tugas akhir.

3.1 Arsitektur Sistem

Pada penelitian tugas akhir ini, digunakan empat buah arsitektur, yaitu: arsitektur *baseline* (subbab 3.1.1), arsitektur sistem POS *tagging* (subbab 3.1.2), arsitektur sistem *parsing* (subbab 3.1.3), dan arsitektur *mixed* (subbab 3.1.4).

3.1.1 Arsitektur Sistem Normal (*Baseline*)

Berikut adalah arsitektur dan perancangan *baseline*

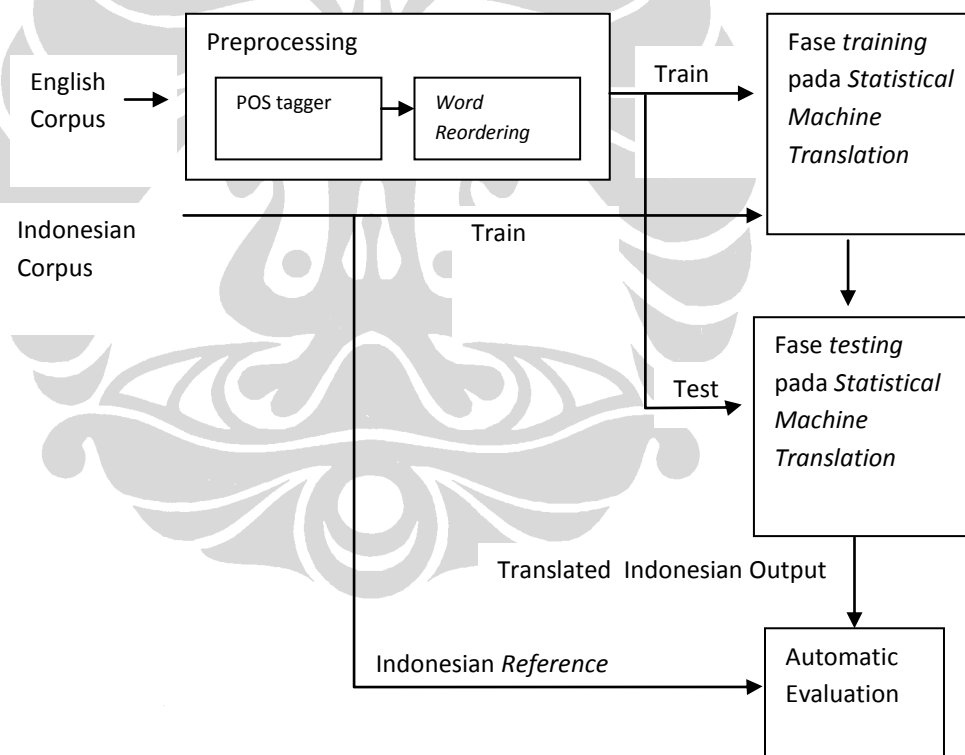


Gambar 3.1 Arsitektur *Baseline*

Pada arsitektur sistem normal, data set korpus paralel Bahasa Inggris dan Bahasa Indonesia langsung dibagi dengan menggunakan metode *K-Fold Cross Validation* (dijelaskan lebih lanjut pada subbab 3.3). Setelah itu dilakukan proses pembuatan *language model* dan *translation model* dengan data set korpus paralel Bahasa Inggris dan Bahasa Indonesia dalam mesin penerjemah statistik. Kemudian *decoder* dari mesin penerjemah statistik melakukan proses penerjemahan pada data set korpus paralel Bahasa Inggris. Lalu hasil penerjemahan dari mesin penerjemah statistik tersebut yang tertulis dalam Bahasa Indonesia dibandingkan dengan data set korpus paralel Bahasa Indonesia yang digunakan sebagai rujukan pada *Automatic Evaluation*.

3.1.2 Arsitektur Sistem POS Tagging (Word Reordering)

Berikut adalah arsitektur dan perancangan sistem POS tagging



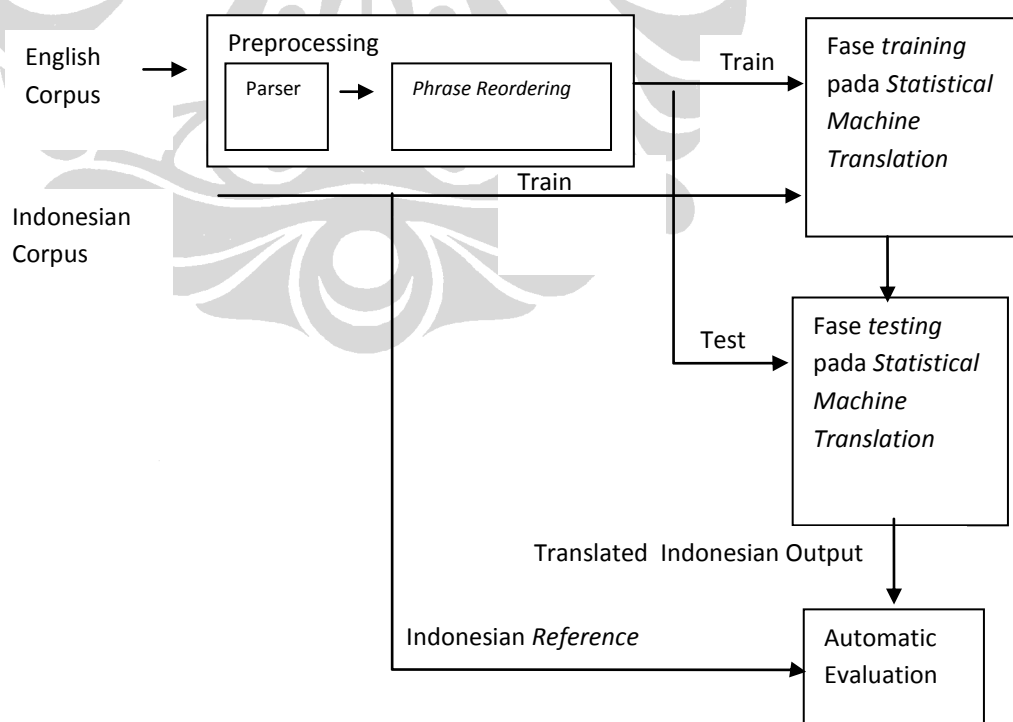
Gambar 3.2 Arsitektur Sistem POS Tagging

Pada arsitektur sistem POS tagging, dilakukan *preprocessing* terlebih dahulu pada data set korpus paralel Bahasa Inggris. Korpus paralel Bahasa Inggris dimasukkan ke dalam POS tagger untuk mendapatkan tanda kelas kata pada setiap kata yang terdapat dalam kalimat. Setelah mendapatkan informasi

tanda kelas kata, dilakukan proses *word reordering* untuk menukar posisi kata dalam kalimat tersebut. Terdapat 7 jenis proses *word reordering* yang dilakukan (dijelaskan lebih lanjut pada subbab 3.4). Setelah dilakukan proses *word reordering*, simbol POS tag yang terdapat pada korpus dihapus. Kemudian data set korpus paralel Bahasa Inggris yang telah melalui *preprocessing* dan Bahasa Indonesia dibagi dengan menggunakan metode *K-Fold Cross Validation* (dijelaskan lebih lanjut pada subbab 3.3). Setelah itu dilakukan proses pembuatan *language model* dan *translation model* dengan data set korpus paralel Bahasa Inggris dan Bahasa Indonesia dalam mesin penerjemah statistik. Kemudian *decoder* dari mesin penerjemah statistik melakukan proses penerjemahan pada data set korpus paralel Bahasa Inggris. Lalu hasil penerjemahan dari mesin penerjemah statistik tersebut yang tertulis dalam Bahasa Indonesia dibandingkan dengan data set korpus paralel Bahasa Indonesia yang digunakan sebagai rujukan pada *Automatic Evaluation*.

3.1.3 Arsitektur Sistem Parsing (*Phrase Reordering*)

Berikut adalah arsitektur dan perancangan sistem *parsing*

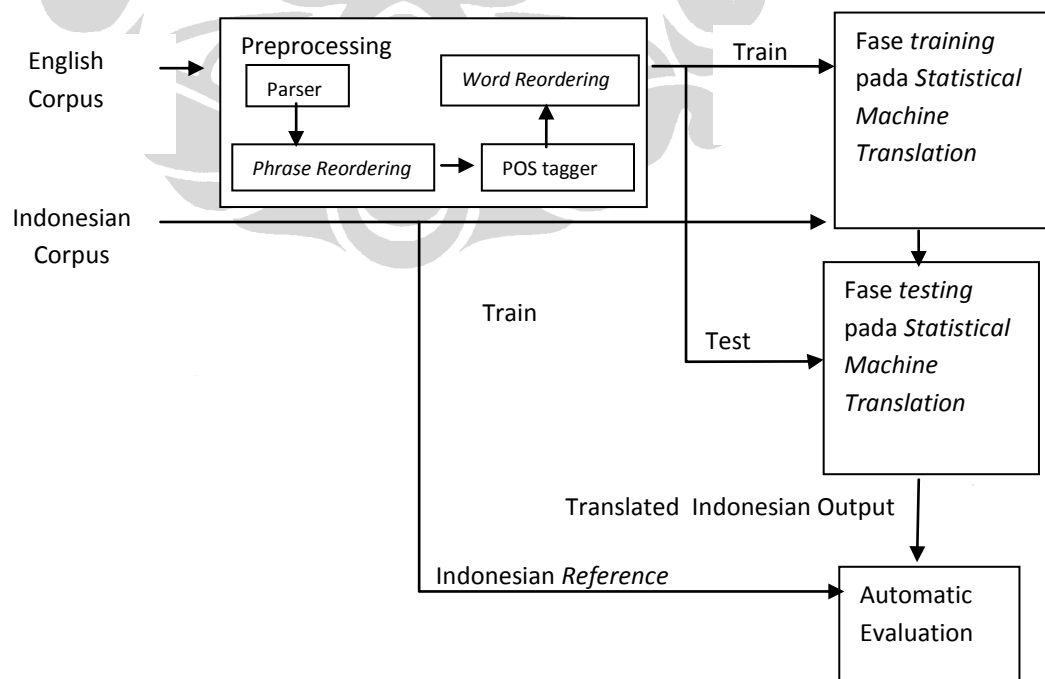


Gambar 3.3 Arsitektur Sistem Parsing

Pada arsitektur sistem *parsing*, dilakukan *preprocessing* terlebih dahulu pada data set korpus paralel Bahasa Inggris. Korpus paralel Bahasa Inggris dimasukkan ke dalam *parser* untuk mendapatkan informasi frase – frase yang terdapat dalam kalimat. Setelah itu, dilakukan beberapa proses *phrase reordering* untuk menukar posisi frase dalam kalimat tersebut. Terdapat 7 jenis proses *phrase reordering* yang dilakukan (dijelaskan lebih lanjut pada subbab 3.5). Kemudian data set korpus paralel Bahasa Inggris yang telah melalui *preprocessing* dan Bahasa Indonesia dibagi dengan menggunakan metode *K-Fold Cross Validation* (dijelaskan lebih lanjut pada subbab 3.3). Setelah itu dilakukan proses pembuatan *language model* dan *translation model* dengan data set korpus paralel Bahasa Inggris dan Bahasa Indonesia dalam mesin penerjemah statistik. Kemudian *decoder* dari mesin penerjemah statistik melakukan proses penerjemahan pada data set korpus paralel Bahasa Inggris. Lalu hasil penerjemahan dari mesin penerjemah statistik tersebut yang tertulis dalam Bahasa Indonesia dibandingkan dengan data set korpus paralel Bahasa Indonesia yang digunakan sebagai rujukan pada *Automatic Evaluation*.

3.1.4 Arsitektur Sistem *Parsing* dan POS *Tagging* (*Mixed*)

Berikut adalah arsitektur dan perancangan *mixed*.



Gambar 3.4 Arsitektur *Mixed*

Pada arsitektur *Mixed*, dilakukan *preprocessing* terlebih dahulu pada data set korpus paralel Bahasa Inggris. Korpus paralel Bahasa Inggris dimasukkan ke dalam *parser* untuk mendapatkan informasi frase – frase yang terdapat dalam kalimat. Setelah itu, dilakukan beberapa proses *phrase reordering* untuk menukar posisi frase dalam kalimat tersebut. Proses *phrase reordering* yang diterapkan adalah ADJP-NNx (dijelaskan lebih lanjut pada subbab 3.5). Setelah proses *phrase reordering*, korpus paralel Bahasa Inggris diproses dengan POS *tagger* dan dilakukan proses *word reordering*. Terdapat 2 macam proses *word reordering* yang diterapkan, yaitu: JJx-NNx dan PRP\$-NNx (dijelaskan lebih lanjut pada subbab 3.4). Setelah dilakukan proses *word reordering*, simbol POS *tag* yang terdapat pada korpus dihapus. Kemudian data set korpus paralel Bahasa Inggris yang telah melalui *preprocessing* dan Bahasa Indonesia dibagi dengan menggunakan metode *K-Fold Cross Validation* (dijelaskan lebih lanjut pada subbab 3.3). Setelah itu dilakukan proses pembuatan *language model* dan *translation model* dengan data set korpus paralel Bahasa Inggris dan Bahasa Indonesia dalam mesin penerjemah statistik. Kemudian *decoder* dari mesin penerjemah statistik melakukan proses penerjemahan pada data set korpus paralel Bahasa Inggris. Lalu hasil penerjemahan dari mesin penerjemah statistik tersebut yang tertulis dalam Bahasa Indonesia dibandingkan dengan data set korpus paralel Bahasa Indonesia yang digunakan sebagai rujukan pada *Automatic Evaluation*.

3.2 Korpus yang Digunakan

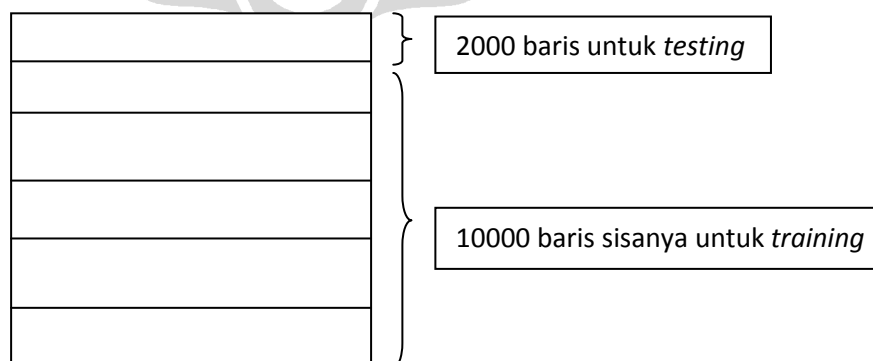
Korpus merupakan koleksi dokumen yang dibaca oleh mesin. Korpus paralel adalah dua buah kumpulan dokumen yang memiliki isi yang sama dan ditulis dalam bahasa yang berbeda. Terdapat dua buah korpus paralel Bahasa Inggris dan Bahasa Indonesia yang digunakan dalam penelitian ini, yaitu: korpus paralel bible dan korpus paralel novel. Korpus paralel bible diperoleh dari penelitian tugas akhir yang dilakukan oleh Eliza Margaretha di bawah bimbingan Hisar Maruli Manurung. Korpus paralel novel diperoleh dari penelitian lab Information Retrieval Fakultas Ilmu Komputer Universitas Indonesia yang dilakukan oleh Aurora Marsye, Desmond Darma Putra, dan Franky di bawah bimbingan Bu Mirna Adriani dalam bidang mesin penerjemah. Sebelum penulis

menggunakan kedua korpus paralel tersebut dalam penelitian, penulis terlebih dahulu melakukan pengecekan tanda baca dan karakter pada korpus paralel. Masing – masing korpus paralel tersebut memiliki 12000 baris.

3.3 K-Fold Cross Validation

Dalam *K-Fold Cross Validation*, data set korpus paralel dibagi menjadi K *subsamples*. Dari K *subsamples*, satu *subsample* digunakan sebagai validasi data untuk *testing*, dan $K - 1$ *subsamples* sisanya digunakan sebagai data *training*. Proses *Cross Validation* ini kemudian diulangi sampai K kali, dengan setiap K *subsamples* tepat digunakan sekali sebagai validasi data. Kemudian dihitung nilai rata – rata dari hasil K percobaan tersebut. Keuntungan dari metode ini adalah bahwa semua bagian dari data set korpus paralel digunakan untuk training dan testing sehingga setiap bagian mendapatkan perlakuan yang sama (Russel & Norvig, 2002).

Dalam penelitian ini penulis menerapkan metode *6-Fold Cross Validation*. Data set korpus paralel yang berjumlah 12000 baris dibagi menjadi 6 buah *fold* yang masing – masing berjumlah 2000 baris. Untuk sebuah eksperimen diambil 1 buah *fold* untuk testing dan 5 buah *fold* sisanya untuk training. Dengan kata lain, dari 12000 baris korpus paralel dibagi menjadi 2000 baris untuk *testing* dan 10000 baris untuk *training*. Sehingga untuk sebuah data set korpus paralel dilakukan 6 kali percobaan dan dari hasil keenam percobaan tersebut diambil nilai rata – ratanya.



Gambar 3.5 Ilustrasi 6-Fold Cross Validation

Gambar 3.5 adalah salah satu ilustrasi *6-Fold Cross Validation*. Dari 12000 baris kalimat, dibagi menjadi 2 bagian yaitu: 2000 baris pertama untuk *testing* dan 10000 baris sisanya untuk *training*.

3.4 Rancangan *Word Reordering*

Penerjemahan kalimat Bahasa Inggris ke Bahasa Indonesia akan melibatkan aturan tata bahasa tertentu. Salah satu aturan tata bahasa yang penting adalah aturan MD-DM (Alwi, 2003). M adalah menerangkan dan D adalah diterangkan. Sebuah frase pada Bahasa Inggris dibentuk dengan aturan menerangkan-diterangkan (MD). Sedangkan sebuah frase pada Bahasa Indonesia frase dibentuk dengan aturan diterangkan-menerangkan (DM). Pola MD berarti kata yang berada di depan menerangkan kata yang berada dibelakangnya, sedangkan pola DM berarti kata yang berada di belakang menerangkan kata yang berada di depannya. Kata yang diterangkan menjadi unsur utama frase dan kata yang menerangkan menjadi unsur penjelas, sehingga penerjemahan kata dari Bahasa Inggris menjadi Bahasa Indonesia akan mengikuti aturan MD-DM. Berikut merupakan contoh penerjemahan yang mengikuti aturan MD- DM:

1. *Intelligent Student* (MD) menjadi siswa pintar (DM)

Kata *Intelligent* yang merupakan kata sifat menerangkan kata *Student* yang merupakan kata benda, sehingga berlaku hukum menerangkan – diterangkan. Kata *Intelligent* diterjemahkan menjadi kata pintar dan kata *Student* diterjemahkan menjadi siswa. Dalam hal ini siswa menjadi kata yang diterangkan dari kata pintar, sehingga berlaku hukum diterangkan - menerangkan.

2. *Branch Office* (MD) menjadi kantor cabang (DM)

Kata *Branch* yang merupakan kata benda menerangkan kata *Office* yang merupakan kata benda, sehingga berlaku hukum menerangkan – diterangkan. Kata *Branch* diterjemahkan menjadi kata cabang dan kata *Office* diterjemahkan menjadi kantor. Dalam hal ini kantor menjadi kata yang diterangkan dari kata cabang, sehingga berlaku hukum diterangkan - menerangkan.

Word reordering adalah penukaran ulang susunan kata dalam sebuah kalimat. Merujuk pada subbab 2.2, *word reordering* dapat meningkatkan kualitas penerjemahan mesin penerjemah statistik. Dalam penelitian ini, penulis mencoba melakukan penukaran ulang susunan kata dalam kalimat Bahasa Inggris agar kalimat tersebut dapat mempunyai susunan kalimat yang semirip mungkin dengan aturan Bahasa Indonesia.

Penulis merancang 3 buah aturan pokok dalam proses *word reordering*, yaitu:

1. *JJx-NNx reordering*

JJx-NNx reordering adalah penukaran urutan kata sifat dengan kata benda. Kata sifat dalam Bahasa Inggris ditandai dengan *tag* /JJ, /JJN, /JJS dan /JJSS. Penulis menyebut kelompok *tag* untuk kata sifat tersebut dengan JJx. Kata benda dalam Bahasa Inggris ditandai dengan *tag* /NN, /NNP, /NNS dan /NNPS. Penulis menyebut kelompok *tag* untuk kata benda tersebut dengan NNx. Apabila pada suatu kalimat terdapat kata sifat yang diikuti dengan kata benda, maka posisi kedua kata tersebut ditukar.

Tabel 3.1 Contoh JJx-NNx Reordering

Inggris	Rudi was reading red book
Inggris (POS <i>tagging</i>)	Rudi/NNP was/VBD reading/VBG <u>red/JJ</u> <u>book/NN</u>
Inggris Reordered	Rudi/NNP was/VBD reading/VBG <u>book/NN</u> <u>red/JJ</u>
Indonesia	Rudi sedang membaca buku merah

Pada Tabel 3.1 dapat dilihat bahwa dilakukan penukaran antara "red/JJ" dengan "book/NN". Bentuk struktur kalimat Bahasa Inggris yang telah ditukar tersebut memiliki bentuk yang lebih mirip dengan struktur kalimat Bahasa Indonesia.

2. PRP\$-NNx *reordering*

PRP\$-NNx *reordering* adalah penukaran urutan kata kepunyaan dengan kata benda. Kata kepunyaan dalam Bahasa Inggris ditandai dengan *tag* /PRP\$. Kata benda dalam Bahasa Inggris ditandai dengan *tag* /NN, /NNP, /NNS dan /NNPS. Penulis menyebut kelompok *tag* untuk kata benda tersebut dengan NNx. Apabila pada suatu kalimat terdapat kata kepunyaan yang diikuti dengan kata benda, maka posisi kedua kata tersebut ditukar.

Tabel 3.2 Contoh PRP\$-NNx *Reordering*

Inggris	Her cat is sleeping
Inggris (POS <i>tagging</i>)	<u>Her/PRP\$</u> <u>cat/NN</u> is/VBZ sleeping/VBG
Inggris Reordered	<u>cat/NN</u> <u>Her/PRP\$</u> is/VBZ sleeping/VBG
Indonesia	Kucingnya sedang tidur

Pada Tabel 3.2 dapat dilihat bahwa dilakukan penukaran antara "Her/PRP\$" dengan "cat/NN". Bentuk struktur kalimat Bahasa Inggris yang telah ditukar tersebut memiliki bentuk yang lebih mirip dengan struktur kalimat Bahasa Indonesia.

3. NNx-NNx *reordering*

NNx-NNx *reordering* adalah penukaran urutan kata benda dengan kata benda. Kata benda dalam Bahasa Inggris ditandai dengan *tag* /NN, /NNP, /NNS dan /NNPS. Penulis menyebut kelompok *tag* untuk kata benda tersebut dengan NNx. Apabila pada suatu kalimat terdapat kata benda yang diikuti dengan kata benda, maka posisi kedua kata tersebut ditukar.

Tabel 3.3 Contoh NNx-NNx *Reordering*

Inggris	The police report was conclusive
Inggris (POS <i>tagging</i>)	The/DT <u>police/NN</u> <u>report/NN</u> was/VBD conclusive/JJ
Inggris Reordered	The/DT <u>report/NN</u> <u>police/NN</u> was/VBD conclusive/JJ
Indonesia	Laporan polisi sudah meyakinkan.

Pada Tabel 3.3 dapat dilihat bahwa dilakukan penukaran antara "report/NN" dengan "police/NN". Bentuk struktur kalimat Bahasa Inggris yang telah ditukar tersebut memiliki bentuk yang lebih mirip dengan struktur kalimat Bahasa Indonesia.

Penulis mendapatkan 3 aturan tersebut dari studi literatur (Alwi, 2003), analisis empiris korpus paralel secara umum dan melalui diskusi dengan pembimbing. Pada tahap implementasi, eksperimen *word reordering* dilakukan dengan mencoba semua kemungkinan dari 3 buah variasi *word reordering* tersebut. Oleh karena itu terdapat 7 buah kombinasi aturan *word reordering*, yaitu:

1. JJx-NNx *reordering*
 2. NNx-NNx *reordering*
 3. PRP\$-NNx *reordering*
 4. JJx-NNx dan NNx-NNx *reordering*
 5. JJx-NNx dan PRP\$-NNx *reordering*
 6. NNx-NNx dan PRP\$-NNx *reordering*
 7. JJx-NNx, NNx-NNx dan PRP\$-NNx *reordering*
- } 1 *reordering*
- } 2 *reordering*
- } 3 *reordering*

Masing - masing aturan tersebut diterapkan pada korpus paralel Bahasa Inggris. Oleh karena itu terdapat 7 buah variasi data set korpus paralel *word reordering* pada korpus paralel Bahasa Inggris bible dan 7 buah variasi data set korpus paralel *word reordering* pada korpus paralel Bahasa Inggris novel.

3.5 Rancangan *Phrase Reordering*

Phrase reordering adalah penukaran ulang susunan frase atau kelompok kata dalam sebuah kalimat. Merujuk pada subbab 2.2 tentang *word and phrase reordering*, *phrase reordering* dapat meningkatkan kualitas penerjemahan mesin penerjemah statistik. Dalam penelitian ini, penulis mencoba melakukan penukaran ulang susunan frase dalam kalimat Bahasa Inggris agar kalimat tersebut dapat mempunyai susunan kalimat yang semirip mungkin dengan aturan Bahasa Indonesia.

Penulis merancang 3 buah aturan pokok dalam proses *phrase reordering*, yaitu:

1. ADJP-NP *reordering*

ADJP-NP *reordering* adalah penukaran urutan frase sifat atau frase adjektival dengan frase benda. ADJP merupakan simbol untuk frase sifat dan NP merupakan simbol untuk frase benda. Apabila pada suatu kalimat terdapat frase sifat yang diikuti dengan frase benda, maka posisi kedua frase tersebut ditukar.

Tabel 3.4 Contoh ADJP-NP *Reordering*

Inggris	he had been <u>asleep</u> <u>only an hour</u> , but he felt like the dead.
<i>Parse Tree</i>	
Inggris reordered	he had been <u>only an hour</u> <u>asleep</u> , but he felt like the dead.
Indonesia	berarti baru <u>satu jam dia tidur</u> , namun seperti mati saja rasanya.

Pada Tabel 3.4 dapat dilihat bahwa dilakukan penukaran antara "asleep" dengan "only an hour asleep". Bentuk struktur kalimat Bahasa Inggris yang telah ditukar tersebut memiliki bentuk yang lebih mirip dengan struktur kalimat Bahasa Indonesia.

2. NP-NP *reordering*

NP-NP *reordering* adalah penukaran urutan frase benda dengan frase benda. NP merupakan simbol untuk frase benda. Apabila pada suatu kalimat terdapat frase benda yang diikuti dengan frase benda, maka posisi kedua frase tersebut ditukar.

Tabel 3.5 Contoh NP-NP *Reordering*

Inggris	tell <u>me</u> <u>your plan</u> .
<i>Parse Tree</i>	<pre> graph TD ROOT[ROOT] --> S[S] S --> VP[VP] S --> NP1[NP] VP --> VB[VB] VP --> NP2[NP] VP --> NP3[NP] VB --> tell[tell] NP2 --> PRP[PRP] PRP --> me[me] NP3 --> PRP\$[PRP\$] PRP\$ --> your[your] NP3 --> NN[NN] NN --> plan[plan] </pre>
Inggris reordered	Tell <u>your plan</u> <u>me</u> .
Indonesia	ceritakan <u>rencanamu</u> <u>padaku</u> .

Pada Tabel 3.5 dapat dilihat bahwa dilakukan penukaran antara "me" dengan "your plan". Bentuk struktur kalimat Bahasa Inggris yang telah ditukar tersebut memiliki bentuk yang lebih mirip dengan struktur kalimat Bahasa Indonesia.

3. ADJP-NNx *reordering*

ADJP-NNx *reordering* adalah penukaran urutan frase sifat dengan kata benda. ADJP merupakan simbol untuk frase sifat dan NNx merupakan simbol untuk kata benda. Apabila pada suatu kalimat terdapat frase sifat yang diikuti dengan kata benda, maka posisi frase sifat dan kata benda tersebut ditukar.

Tabel 3.6 Contoh ADJP-NNx Reordering

Inggris	the teacher seemed an <u>exceptionally careful</u> man.
Parse Tree	<pre> graph TD ROOT[ROOT] --> S[S] S --> NP1[NP] S --> VP[VP] NP1 --> DT1[DT] NP1 --> NN1[NN] DT1 --> the[the] NN1 --> teacher[teacher] VP --> VBD[VBD] VP --> NP2[NP] VBD --> seemed[seemed] NP2 --> DT2[DT] NP2 --> ADJP[ADJP] NP2 --> NN2[NN] DT2 --> an[an] ADJP --> RB[RB] ADJP --> JJ[JJ] RB --> exceptionally[exceptionally] JJ --> careful[careful] NN2 --> man[man] </pre>
Inggris reordered	the teacher seemed an <u>man</u> <u>exceptionally careful</u> .
Indonesia	guru tampaknya <u>orang yang betul-betul berhati - hati</u> .

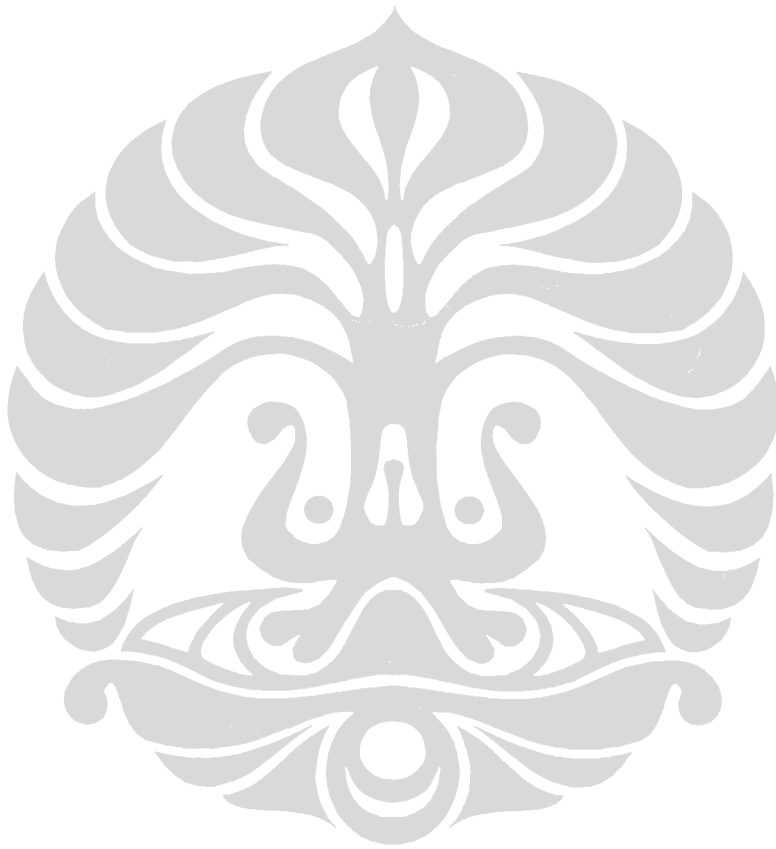
Pada Tabel 3.6 dapat dilihat bahwa dilakukan penukaran antara "exceptionally careful" dengan "man". Bentuk struktur kalimat Bahasa Inggris yang telah ditukar tersebut memiliki bentuk yang lebih mirip dengan struktur kalimat Bahasa Indonesia.

Penulis mendapatkan 3 aturan tersebut dari analisis empiris korpus secara umum dan melalui diskusi dengan pembimbing. Pada tahap implementasi, eksperimen *phrase reordering* dilakukan dengan mencoba semua kemungkinan dari 3 buah variasi *phrase reordering* tersebut. Sehingga terdapat 7 buah kombinasi aturan *phrase reordering*, yaitu:

1. ADJP-NNx *reordering*
 2. NP-NP *reordering*
 3. ADJP-NP *reordering*
 4. ADJP-NNx dan NP-NP *reordering*
 5. ADJP-NNx dan ADJP-NP *reordering*
 6. ADJP-NP dan NP-NP *reordering*
- } 1 *reordering*
 } 2 *reordering*

7. ADJP-NN_x, NP-NP *reordering*, dan ADJP-NP *reordering* } 3 *reordering*

Masing - masing aturan *phrase reordering* tersebut diterapkan pada korpus paralel Bahasa Inggris. Oleh karena itu terdapat 7 buah variasi data set korpus paralel *phrase reordering* pada korpus paralel Bahasa Inggris bible dan 7 buah variasi data set korpus paralel *phrase reordering* pada korpus paralel Bahasa Inggris novel.



BAB 4 IMPLEMENTASI

Bab ini menjelaskan tentang proses yang dilalui pada saat implementasi penelitian. Pembahasan dimulai dari persiapan awal dokumen (subbab 4.1), Stanford POS Tagger (subbab 4.2), *word reordering* (subbab 4.3), *cleaning* dokumen (subbab 4.4), Stanford Parser (subbab 4.5), *phrase reordering* (subbab 4.6), MOSES (subbab 4.7), dan *evaluation tools* (subbab 4.8). Subbab pada bab ini penulis susun dalam urutan sistematis, sesuai dengan alur yang penulis lalui selama masa tugas akhir. Proses persiapan awal dokumen, *word reordering* dan *cleaning* dokumen menggunakan bahasa pemrograman Perl. Proses *phrase reordering* menggunakan bahasa pemrograman Java. Untuk proses evaluasi penulis menggunakan *tools* yang sudah tersedia dalam bahasa Perl.

4.1 Persiapan Awal Dokumen

Proses persiapan awal dokumen ini dilakukan untuk sebelum dilakukan proses penerjemahan. Proses ini dilakukan untuk memberi penambahan titik pada setiap akhir kalimat. Hal ini dikarenakan masih ada beberapa kalimat yang belum memiliki titik pada akhir kalimat yang dapat menimbulkan *error* pada proses *tagging* dengan Stanford POS Tagger dan *parsing* dengan Stanford Parser.

```
Function main() {
  file <- openFile(getPOSTaggedEnglishFile);

  for each lineSentence in file{
    if lineSentence not end with '.'
      lineSentence <- add '.' on end lineSentence
  }

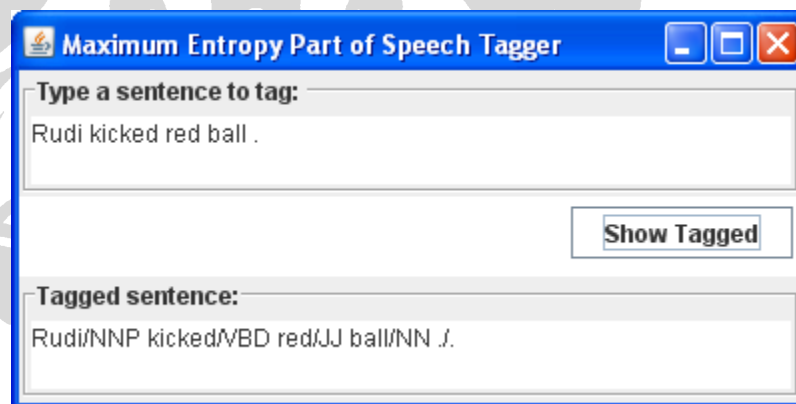
  print to file_out <- newSentences
}
```

Gambar 4.1 Pseudocode Persiapan Awal Dokumen

4.2 Stanford POS Tagger

Stanford POS Tagger adalah software yang membaca teks dalam format bahasa tertentu seperti Inggris, Arab, dan Cina dan memberikan kelas kata pada

setiap kata seperti *noun*, *verb*, *adjective* dan kelas kata lainnya (Stanford University, 2008). Versi yang digunakan dalam penelitian ini adalah Stanford POS Tagger versi 1.6. Stanford POS Tagger membutuhkan platform Java 1.5 dan 1 GB memori. Stanford POS Tagger dapat dijalankan secara interaktif dengan user dan dengan cara otomatis. Proses *tagging* secara interaktif dengan *Graphical User Interface* (GUI) dapat dijalankan dengan memanggil "java -jar stanford-postagger.jar". Sedangkan cara menjalankan secara otomatis adalah dengan memanggil "java -mx1000m -classpath stanford-postagger.jar edu.stanford.nlp.tagger.maxent.MaxentTagger -model models/left3words-wsj-0-18.tagger -textFile input.txt > output.txt". Proses *tagging* dilakukan secara otomatis dengan cara membaca setiap baris pada berkas "input.txt" dan menuliskan hasil *tagging* tersebut pada "output.txt". Stanford POS Tagger versi 1.6 dapat didownload di <http://nlp.stanford.edu/software/stanford-postagger-full-2008-09-28.tar.gz>. Berikut adalah contoh tampilan Stanford POS Tagger dengan kalimat "Rudi kicked red and blue ball".



Gambar 4.2 Stanford POS Tagger

Rudi diberi simbol kelas kata NNP yang berarti *proper noun*, kicked diberi simbol kelas kata VBD yang berarti *verb past tense*, Blue diberi simbol kelas kata JJ yang berarti *adjective*, And diberi simbol kelas kata CC yang berarti *coordinating conjunction*, Red diberi simbol kelas kata JJ yang berarti *adjective*, dan Ball diberi simbol kelas kata NN yang berarti *noun*.

4.3 Word Reordering

Sesuai dengan penjelasan pada subbab 3.4, akan dilakukan 7 macam variasi proses *word reordering*. Berikut ini merupakan *pseudocode* yang digunakan penulis dalam proses *word reordering*:

```

Function main() {
    file <- openFile(getPOSTaggedEnglishFile)
    lineNo <- 1;
    for each lineSentence in file
    {
        array_word <- split(/\s+/, lineSentence);
        append to array_newSentenceCollection <-
1wordReordering(kelasKatal, array_word) or
2wordReordering(kelasKatal, kelasKata2, array_word) or
3wordReordering(kelasKatal, kelasKata2, kelasKata3,
array_word);

        lineNo++;
    }
    print to fileout_reordered <- array_newSentenceCollection;
    print to fileout_totalReordering <- countReordering;
}

function 1wordReordering(kelasKatal, array_word){
    for each word in array_word{
        if word is kelasKatal and nextWord is kata benda
            {swap(word,nextWord); countReordering++;}

        if (swapOccured) print to fileout_lineReordered <-lineNo;
    }
}

function 2wordReordering(kelasKatal, kelasKata2, array_word){
    for each word in array_word{
        if (word is (kelasKatal or kelasKata2) and nextWord is
kata benda
            {swap(word,nextWord); countReordering++;}

        if (swapOccured) print to fileout_lineReordered <-lineNo;
    }
}

function 3wordReordering(kelasKatal, kelasKata2, kelasKata3,
array_word){
    for each word in array_word{
        if (word is kelasKatal or kelasKata2 or kelasKata3) and
nextWord is kata benda
            {swap(word,nextWord); countReordering++;}

        if (swapOccured) print to fileout_lineReordered <-lineNo;
    }
}

```

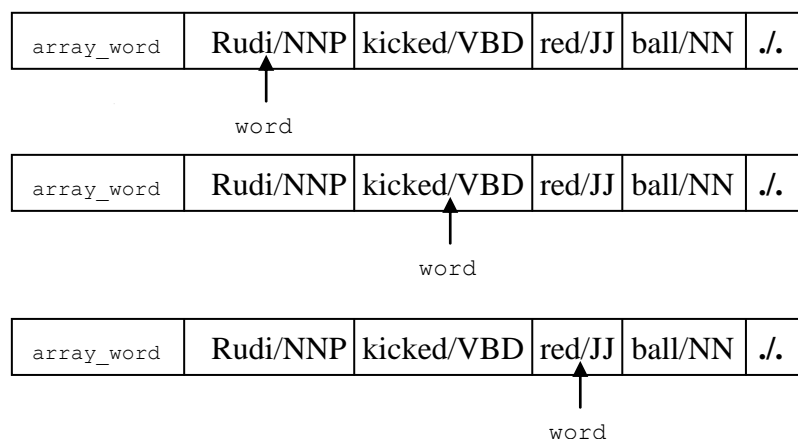
Gambar 4.3 *Pseudocode Word Reordering*

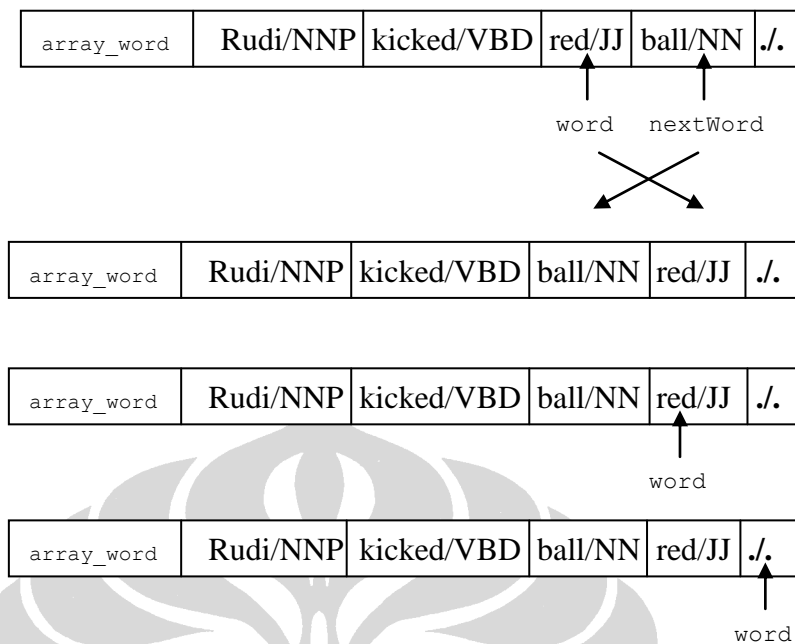
Proses awal dari *pseudocode* tersebut adalah membaca berkas korpus paralel Bahasa Inggris yang telah melalui tahap POS *tagging*. Setiap kata dari

berkas korpus paralel Bahasa Inggris tersebut telah mempunyai informasi kelas katanya masing - masing. Berkas Bahasa Inggris tersebut dibaca per baris atau per kalimat. Kemudian dilakukan proses tokenisasi pada kalimat dengan pemisah spasi dan menyimpannya dalam variabel `array_word`. Setelah itu dilakukan proses *word reordering* pada kumpulan kata dari kalimat tersebut sesuai dengan aturan yang ingin diterapkan (merujuk pada subbab 3.4 terdapat 7 aturan *word reordering*). Hasil proses `wordReordering` tersebut disimpan dalam variabel `array_newSentenceCollection`. Setelah semua baris atau kalimat pada berkas input dibaca, maka hasil proses `wordReordering` dan jumlah proses `wordReordering` yang terjadi disimpan pada berkas output.

Proses `wordReordering` digunakan untuk melakukan penyusunan ulang kata pada sebuah kalimat. Proses tersebut dimulai dengan mengecek kelas kata dari setiap kata dalam suatu kalimat dengan variasi aturan yang ingin diterapkan. Sebagai contoh, apabila akan diterapkan proses penukaran antara kata sifat dengan kata benda (*1WordReordering*), maka dilakukan pencarian kata sifat mulai dari awal kalimat sampai akhir kalimat. Jika ditemukan suatu kata sifat, maka dilakukan pengecekan sebuah kata berikutnya dari kata sifat tersebut. Apabila kata berikutnya merupakan kata benda, maka dilakukan proses penukaran kata. Variabel `countReordering1` akan bertambah dan nomor baris kalimat yang mengalami proses penukaran tersebut akan disimpan pada sebuah berkas.

Misalkan terdapat sebuah kalimat "Rudi kicked red ball." Maka proses penukaran kata sifat dengan kata benda adalah:





Gambar 4.4 Ilustrasi JJx-NNx pada Kalimat "Rudi kicked red ball."

Pada ilustrasi di atas kata sifat "red" yang mempunyai *tag* /JJ ditukar dengan kata benda "ball" yang mempunyai *tag* /NN.

4.4 Cleaning Dokumen

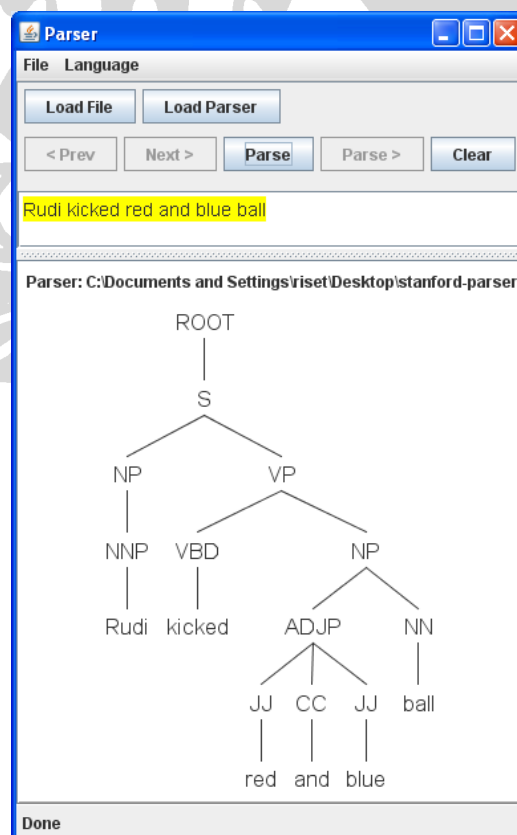
Proses *Cleaning* Dokumen merupakan proses penghilangan simbol POS *tag* setelah korpus paralel melalui tahap *word reordering*. Input yang diperlukan pada proses ini adalah satu buah berkas yang berisi kalimat Bahasa Inggris yang masih memiliki POS *tag*. Output dari proses ini digunakan untuk proses penerjemahan dengan MOSES.

```
Function main() {
  file <- openFile(getPOSTaggedEnglishFile);
  for each lineSentence in file{
    for each word in lineSentence
      newSentences<- removeAllCharacter from index character
      '/' until wordlength;
    }
  print to file_out <- newSentences;
}
```

Gambar 4.5 Pseudocode Cleaning Dokumen

4.5 Stanford Parser

Stanford Parser adalah *software* yang digunakan untuk melakukan *parsing* kalimat Bahasa Inggris. Stanford Parser ini digunakan untuk menentukan kelompok kata atau frase dalam suatu kalimat. Versi original dari *parser* ini dibuat oleh Dan Klein dengan bantuan kode dan *linguistic grammar* yang dikembangkan oleh Christopher Manning (Stanford University, 2008). Versi yang digunakan dalam penelitian ini adalah Stanford Parser versi 1.6.1. Stanford Parser tersebut membutuhkan platform Java 1.5 dan 1 GB memori. Stanford Parser dapat dijalankan dengan cara memanggil "java -jar stanford-parser.jar" pada command prompt. Stanford Parser menyediakan beberapa aturan *grammar* yang dapat digunakan dalam *parser* seperti: *arabicFactored*, *chineseFactored*, dan *englishPCFG*. Stanford Parser versi 1.6.1 dapat didownload di <http://nlp.stanford.edu/software/stanford-parser-2008-10-26.tgz>. Contoh tampilan Stanford Parser dengan menggunakan *englishPCFG.ser.gz* dan kalimat "Rudi kicked red and blue ball".



Gambar 4.6 Stanford Parser

“Rudi” diberi simbol kelas kata NNP yang berarti *proper noun*, “kicked” diberi simbol kelas kata VBD yang berarti *verb past tense*, “blue” diberi simbol kelas kata JJ yang berarti *adjective*, “and” diberi simbol kelas kata CC yang berarti *coordinating conjunction*, “red” diberi simbol kelas kata JJ yang berarti *adjective*, dan “ball” diberi simbol kelas kata NN yang berarti *noun*. Kata “Rudi” membentuk NP yang berarti *noun phrase*. Kata “red and blue” membentuk ADJP yang berarti *adjective phrase*. Kata “red and blue” dan kata “ball” membentuk NP yang berarti *noun phrase*. Kata “kicked” dan kata “red and blue ball” membentuk VP yang berarti *verb phrase*. Kata “Rudi” dan “kicked red and blue ball” membentuk S yang berarti *sentence*.

4.6 Phrase Reordering

Sesuai dengan penjelasan pada subbab 3.5, akan dilakukan 7 macam variasi proses *phrase reordering*. Berikut ini merupakan *pseudocode* yang digunakan penulis dalam proses *phrase reordering*:

```

Function main() {
  file <- openFile(englishFile)
  for each lineSentence in file{
    ParseTree <-- convertSentenceToParseTree(lineSentence);
    result <-- PhraseReordering (P1, P2, parseTree);
    print to fileout_reordered <-- result;
  }
  print to fileout_totalReordering <- countReordering;
}

Function phraseReordering (P1, P2, parseTree){
  list_P1Tree <-- BFSSearchandCollectPhrase(P1, parseTree);
  for each P1Tree in list_P1Tree{
    if list_blacklistedTree is not contain P1Tree{
      P1Parent <-- findParent(P1Tree);
      list_P1SiblingTree <-- getAllChildrenTree(P1Parent)
      indexP1 <- getIndexFromTree(P1Tree, P1Parent);
      //searching P2 in right sibling P1Tree
      for (int i = indexP1+1; i<list_P1SiblingTree.size();
i++){
        siblingTree <-- list_P1SiblingTree.get(i);
        if(siblingTree is P2){
          swapTree(P1Tree, siblingTree);
          blackListedTree.add(siblingTree);
          countReordering++;
          print to fileout_lineReordered <-lineNo;
          break;
        }
      }
    }
  }
}
}

```

```

Function BFSSearchandCollectPhrase(phrase, parseTree){
  list_tempTree.add(parseTree);
  while (list_tempTree is not empty){
    tree_curent <- list_tempTree.remove();
    list_childrenTree <- getAllChildrenTree(tree_current);
    for each childTree in list_childrenTree
      list_tempTree.add(childTree);

    if(tree_current.value() is phrase)
      list_result.add(tree_current);
  }
  return list_result
}

```

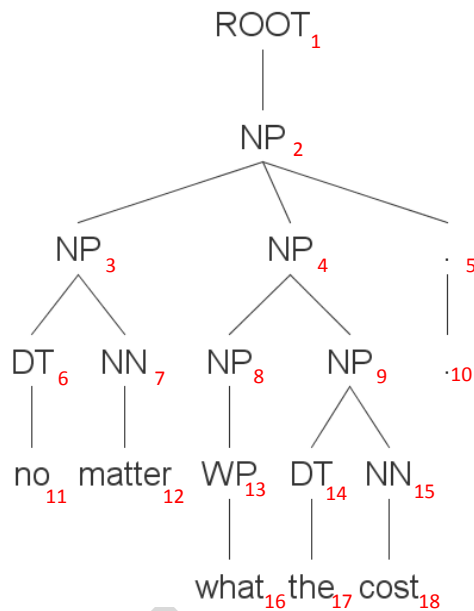
Gambar 4.7 Pseudocode Phrase Reordering

Proses awal dari *pseudocode* tersebut adalah membaca berkas korpus paralel Bahasa Inggris. Berkas *input* Bahasa Inggris tersebut dibaca per baris atau per kalimat. Kemudian setiap baris kalimat tersebut diubah menjadi sebuah *parse tree* dengan menggunakan *library* dari Stanford Parser. Setelah itu dilakukan proses *phraseReordering* pada kalimat tersebut. Hasil proses *phraseReordering* tersebut ditulis ke dalam sebuah berkas *output*. Setelah itu proses dilanjutkan dengan membaca baris berkas *input* berikutnya. Setelah semua baris atau kalimat pada berkas *input* dibaca, maka jumlah proses *phraseReordering* yang terjadi disimpan ke dalam sebuah berkas *output*.

Fungsi *phraseReordering* pada *pseudocode* di atas digunakan untuk melakukan penyusunan ulang frase pada sebuah kalimat. Proses tersebut dimulai dengan pencarian dan penyimpanan suatu frase tertentu dalam sebuah *parse tree*.

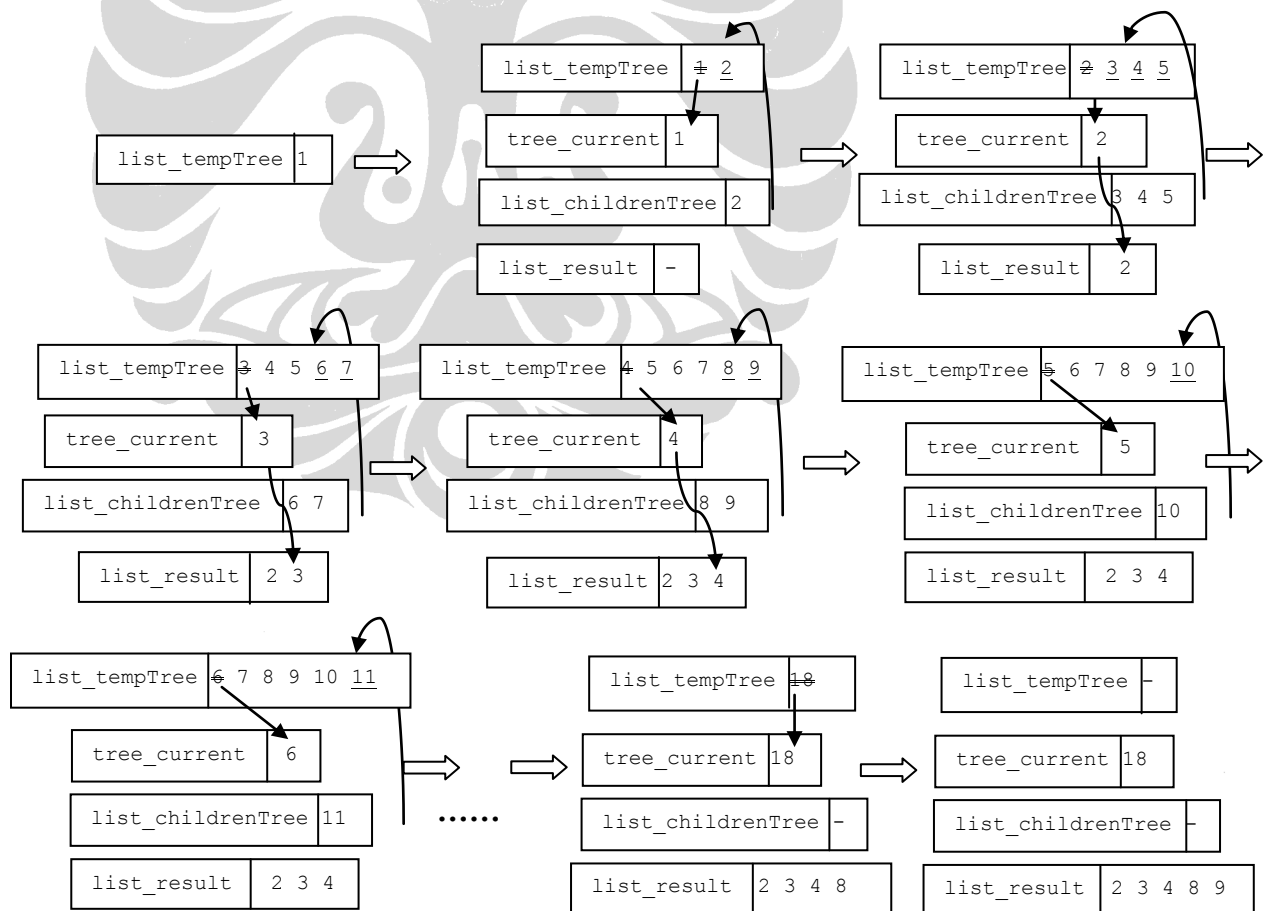
Misalkan akan dilakukan penukaran NP-NP (frase benda dengan frase benda) dalam kalimat "no matter what the cost.", maka prosesnya adalah sebagai berikut:

1. Membuat *parse tree* dari kalimat "no matter what the cost." Gambar 4.8 adalah *parse tree* dari kalimat tersebut.



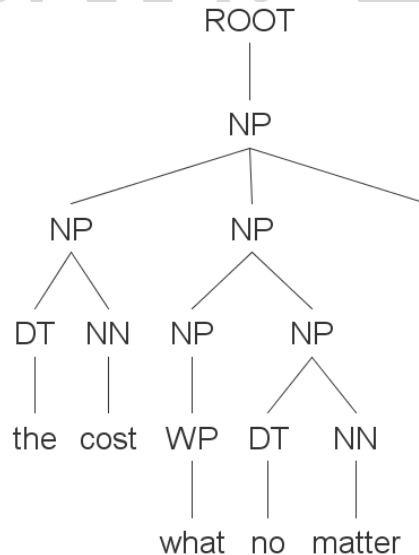
Gambar 4.8 Parse Tree Kalimat "no matter what the cost."

- Menjalankan BFSSearchandCollectPhrase, mencari semua jenis frase NP dalam kalimat. Proses pencarian dilakukan secara BFS, dimulai dari 1 sampai dengan 18.



Gambar 4.9 Ilustrasi dari BFSSearchandCollectPhrase

3. Setelah didapatkan semua *tree* NP dalam suatu kalimat, maka dilakukan pencarian *right sibling* untuk setiap isi dari `list_result` tersebut. Apabila *tree* NP tersebut mempunyai *right sibling* yang bersimbolkan NP (untuk menghindari kerancuan, *right sibling* yang bersimbol NP ini dinamakan NP*), maka dilakukan penukaran antara *tree* NP dengan *tree* NP*. Jumlah maksimal penukaran yang terjadi pada suatu *tree* adalah 1 kali. Setelah NP* tersebut mengalami proses penukaran, NP* tersebut dimasukkan ke dalam `blackListedTree`. Hal ini bertujuan untuk membedakan *tree* NP yang sudah mengalami proses penukaran dengan *tree* NP yang belum mengalami proses penukaran. Pada contoh kalimat di atas, proses penukaran frase yang terjadi adalah: penukaran *tree* 3 dengan *tree* 4 dan penukaran *tree* 8 dengan *tree* 9. Proses pencarian *right sibling* untuk *tree* 4 tidak dilakukan, karena *tree* 4 sudah dimasukkan terlebih dahulu ke dalam `blackListedTree` saat *tree* 3 ditukar dengan *tree* 4. Demikian juga untuk *tree* 9, sehingga hasil kalimat tersebut setelah mengalami penukaran frase NP dengan NP adalah: "the cost what no matter."



Gambar 4.10 Parse Tree Kalimat "the cost what no matter."

Gambar 4.10 adalah gambar *parse tree* pada kalimat "the cost what no matter."

4.7 MOSES, SRILM, GIZA++

MOSES merupakan sebuah *open source software* sistem mesin penerjemah statistik yang dapat digunakan untuk training dan testing pada dokumen pasangan bahasa yang paralel. Untuk menjalankan MOSES diperlukan software lain yang bernama SRILM dan GIZA++. MOSES dapat didownload di <http://mosesdecoder.sourceforge.net/download.php>.

Software MOSES, SRILM, dan GIZA++ sudah terinstallasi dan terintegrasi pada salah satu komputer di LAB-Information Retrieval, sehingga penulis dapat langsung menggunakan software tersebut.

SRILM merupakan sebuah *software* yang digunakan untuk membangun dan menerapkan *language model* statistik (LMs). SRILM biasanya digunakan untuk keperluan *speech recognition*, *statistical tagging*, dan mesin penerjemah. SRILM dikembangkan oleh *SRI Speech Technology and Research Laboratory* sejak tahun 1995. SRILM digunakan MOSES untuk membuat *language model* dari data training. SRILM dapat didownload di <http://www.speech.sri.com/projects/srilm/download.html>. Penulis menggunakan SRILM versi 1.5.6 yang merupakan versi terakhir ketika eksperimen dilaksanakan.

GIZA++ merupakan perkembangan dari GIZA. GIZA++ adalah sebuah software yang digunakan MOSES untuk melakukan pencocokan kata atau *word alignment*. GIZA++ dirancang oleh Franz Josef Och. GIZA++ dapat didownload di <http://www.fjoch.com/GIZA++.2003-09-30.tar.gz>. Penulis menggunakan GIZA++ versi 1.0.2 yang merupakan versi terakhir ketika eksperimen dilaksanakan.

Proses *training* pada MOSES dilakukan dengan menjalankan sebuah *script* dengan format `.sh` yang terletak di struktur direktori `/MOSES/moses/program/scripts/`. Berikut adalah contoh *script* training bible-experiment1.sh yang digunakan pada penelitian ini:

```

#!/bin/sh
export PATH=$PATH:/home/riset/MOSES/moses/bin/srilm/bin
export MAN_PATH=/home/riset/MOSES/moses/bin/srilm/man
export SCRIPTS_ROOTDIR=/home/riset/MOSES/moses/bin/scripts-
20081205-1518
export WORKING_DIR=/home/riset/MOSES/workspace
export CORPUS=10000-b-baseline-training-experiment1
export LANG_SOURCE=en
echo "Source language: " $LANG_SOURCE
export LANG_TARGET=id
echo "Target language: " $LANG_TARGET
echo
echo "clean-corpus-n.perl ..."
echo
$SCRIPTS_ROOTDIR/training/clean-corpus-n.perl
$WORKING_DIR/training-data/$CORPUS $LANG_SOURCE $LANG_TARGET
$WORKING_DIR/training-data/$CORPUS.clean 1 100
# ngram
echo
echo "ngram-count ..."
echo
/home/riset/MOSES/moses/bin/srilm/bin/i686/ngram-count -sort \
-order 3 \
-interpolate \
-text $WORKING_DIR/training-data/$CORPUS.clean.$LANG_TARGET \
-lm $WORKING_DIR/lm/$CORPUS.clean.$LANG_TARGET.srilm
# we put each language model on its own directory
mkdir -p $WORKING_DIR/lm/$CORPUS
/home/riset/MOSES/moses/bin/srilm/bin/i686/ngram-count -sort
-order 3 -interpolate -text \
$WORKING_DIR/training-data/$CORPUS.clean.$LANG_TARGET -lm \
$WORKING_DIR/lm/$CORPUS.clean.$LANG_TARGET.srilm
# train
echo
echo "TRAINING"
echo "train-factored-phrase-model.perl ..."
echo
# we put each translation model into different directory
mkdir -p $WORKING_DIR/model/$CORPUS
$SCRIPTS_ROOTDIR/training/train-factored-phrase-model.perl \
-scripts-root-dir $SCRIPTS_ROOTDIR \
-root-dir $WORKING_DIR/model/$CORPUS \
--corpus $WORKING_DIR/training-data/$CORPUS \
--f $LANG_SOURCE --e $LANG_TARGET \
-alignment grow-diag-final-and \
-reordering msd-bidirectional-fe \
--lm 0:3:$WORKING_DIR/lm/$CORPUS.clean.$LANG_TARGET.srilm:0

```

Gambar 4.11 Script *Training* pada MOSES

Dari gambar 4.11, dapat dilihat bahwa ada beberapa hal penting dari *script* pelatihan yang digunakan, antara lain:

1. Nama korpus paralel pelatihan yang digunakan dalam proses training untuk Bahasa Inggris dan Bahasa Indonesia adalah sama, yang berbeda

adalah format berkasnya saja. Pada *script* di atas, format berkas korpus Bahasa Inggris adalah en (dimana LANG_SOURCE=en) dan format berkas korpus Bahasa Indonesia adalah id (dimana LANG_TARGET=id). Untuk seluruh *script* pelatihan yang digunakan, korpus yang dilatih memiliki nama yang mencerminkan jumlah kalimat yang digunakan, korpus paralel yang digunakan, dan urutan fold korpus tersebut. Pada contoh *script* di atas, korpus paralel yang digunakan berasal dari bible dengan jumlah kalimat 10000, dan merupakan fold pertama dari arsitektur *baseline*.

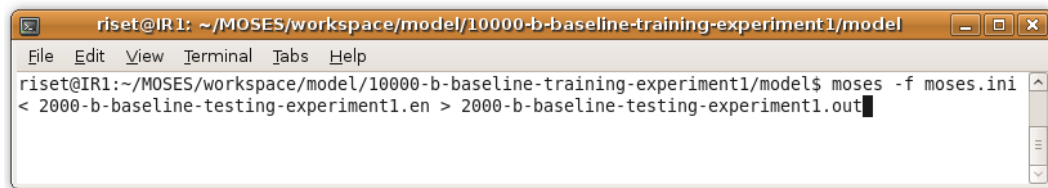
2. Kedua berkas korpus paralel tersebut harus ditempatkan di MOSES/Workspace/training-data.
3. Pada penelitian ini, model bahasa yang digunakan adalah model bahasa trigram (3-gram) sehingga parameter order akan bernilai 3 (order 3). Model bahasa trigram yang merupakan model bahasa *default* yang digunakan oleh MOSES. Parameter `lm 0:3:...` berarti bahwa model bahasa yang dibuat akan memuat isi berupa faktor 0 (yaitu bentuk *surface* dari kata-kata) dari korpus pelatihan bahasa target (Inggris) dalam bentuk 1-gram (1 kata), 2-gram (frase 2 kata) dan 3-gram (frase 3 kata).

Pada penelitian ini, banyaknya *script* yang dibuat sesuai dengan banyaknya *training* yang dilakukan yaitu sebanyak 204 buah *script* (12 *script* untuk arsitektur *baseline*, 84 *script* untuk arsitektur POS *tagging*, 84 *script* untuk arsitektur *parsing*, dan 24 *script* untuk arsitektur *mixed*). Antara *script* yang satu dengan *script* lainnya yang berbeda adalah nama korpus yang digunakan (CORPUS=...), sedangkan untuk parameter order dan `lm 0:3` adalah sama.

Setelah menjalankan proses *training*, proses *testing* dilaksanakan dengan menjalankan berkas yang bernama `moses.ini` yang dihasilkan dari proses *training*. Berkas `moses.ini` dapat ditemukan pada “MOSES/workspace/model/<nama training korpus>/model/”. Proses testing dapat dilakukan dengan 2 cara, yaitu: testing secara interaktif dengan user dan testing secara otomatis. Proses testing secara interaktif dapat dilakukan dengan menjalankan perintah “`moses -f moses.ini`” pada terminal linux. Sedangkan proses

testing secara otomatis dapat dilakukan dengan menjalankan perintah ”moses -f moses.ini < datatest.in > hasiltranslate.out”. Proses penerjemahan dilakukan secara otomatis pada setiap baris pada berkas *input* “datatest.in” dan menuliskan hasilnya pada berkas *output* ”hasiltranslate.out”.

Berikut adalah contoh cara menjalankan testing secara otomatis.



```

riset@IR1: ~/MOSES/workspace/model/10000-b-baseline-training-experiment1/model
File Edit View Terminal Tabs Help
riset@IR1:~/MOSES/workspace/model/10000-b-baseline-training-experiment1/model$ moses -f moses.ini
< 2000-b-baseline-testing-experiment1.en > 2000-b-baseline-testing-experiment1.out
  
```

Gambar 4.12 Contoh menjalankan *testing* secara otomatis

Pada Gambar 4.12, berkas `moses.ini` terletak di `MOSES/workspace/model/10000-b-baseline-training-experiment1/model`. Berkas konfigurasi `moses.ini` dijalankan dengan menggunakan perintah `moses -f moses.ini`. *Input* yang diterima oleh MOSES adalah sebuah berkas korpus *testing* dalam Bahasa Inggris (`2000-b-baseline-testing-experiment1.en`). Hasil terjemahan dari berkas *input* dalam Bahasa Indonesia ditulis ke dalam sebuah berkas *output* (`2000-b-baseline-testing-experiment1.out`).

4.8 Evaluation Tools

`Mteval-v11b.pl` merupakan sebuah *tools* dalam bahasa Perl yang digunakan untuk menjalankan proses *automatic evaluation* dengan standar penilaian BLEU dan NIST. Program `Mteval-v11b.pl` dibuat oleh seorang peneliti IBM yang bernama Kishore Papineni. Program `Mteval-v11b.pl` dapat didownload di <ftp://jaguar.ncsl.nist.gov/mt/resources/mteval-v11b.pl>.

Untuk melakukan proses evaluasi secara otomatis dengan `Mteval-v11b.pl`, berkas *input* harus disimpan dalam format SGM. Berkas *input* dengan format SGM terbagi ke dalam tiga jenis:

- a. *Source file*. *Source file* merupakan berkas yang berisi kalimat dalam bahasa sumber atau Bahasa Inggris. Berkas ini memiliki format khusus sebagai berikut:

```
<srcset setid="example_set" srclang="English"
trglang="Indonesian">
<DOC docid="test1">
<p>
<seg id="1"> ... </seg>
</p>
<p>
<seg id="2"> ... </seg>
</p>
.
.
.
<p>
<seg id="n"> ... </seg>
</p>
</DOC>
</srcset>
```

Gambar 4.13 Struktur Source File

```
Function main() {
file <- openFile(EnglishFile)

print to file_out ("<srcset setid=\"example_set\"srclang=\"English\"
trglang=\"Indonesian\">\n");

print to file_out ("<DOC docid=\"id\">\n");

for each lineSentence in file
{
chomp(lineSentence);
lineSentence <- "<p>\n<seg id=\"counter\">".lineSentence."</seg>\n</p>\n";

print to file_out (lineSentence);
counter++;
}

print to file_out ("</DOC>\n</srcset>");
}
```

Gambar 4.14 Pseudocode Mengubah File Menjadi Source File

- b. *Test file*. *Test file* merupakan berkas yang berisi kalimat dalam bahasa tujuan atau Bahasa Indonesia. Berkas ini berisi hasil terjemahan dari MOSES. Berkas ini memiliki format khusus sebagai berikut:

```
<tstset setid="example_set" srclang="English"
trglang="Indonesian">
<DOC docid="test1" sysid="system">
<p>
<seg id=1> ... </seg>
</p>
<p>
<seg id=2> ... </seg>
</p>
.
.
.
<p>
<seg id=n> ... </seg>
```

```

</p>
</DOC>
</tstset>

```

Gambar 4.15 Struktur Test File

```

Function main() {
file <- openFile(getTranslatedIndonesianFile)
print to file_out ("<tstset setid=\"example_set\" srclang=\"English\"
trglang=\"Indonesian\">\n");
print to file_out ("<DOC docid=\"id\" sysid=\"system\">\n");
for each lineSentence in file
{
  chomp(lineSentence);
  lineSentence <- "<p>\n<seg id=\"counter\">".lineSentence."</seg>\n</p>\n";
  print to file_out (lineSentence);
  counter++;
}
print to file_out ("</DOC>\n</tstset>");
}

```

Gambar 4.16 Pseudocode Mengubah File Menjadi Test File

c. *Reference file*. *Reference file* merupakan berkas yang berisi kalimat dalam bahasa tujuan atau Bahasa Indonesia. Berkas ini merupakan hasil terjemahan yang dilakukan oleh manusia. *Reference file* digunakan sebagai pembandingan sebgus apa kualitas terjemahan yang dihasilkan oleh Moses di dalam test set. Berkas ini memiliki format khusus sebagai berikut:

```

<refset setid="example_set" srclang="English"
trglang="Indonesian">
< DOC docid="doc1" sysid="ref1">
<p>
<seg id=1> ... </seg>
</p>
<p>
<seg id=2> ... </seg>
</p>
.
.
.
<p>
<seg id=n> ... </seg>
</p>
</DOC>
</refset>

```

Gambar 4.17 Struktur Reference File

```

Function main() {
file <- openFile(getReferencedIndonesianFile)
print to file_out "<refset setid=\"example_set\" srclang=\"English\"
trglang=\"Indonesian\">\n ";
print to file_out ("<DOC docid=\"id\" sysid=\"reference\">\n");
for each lineSentence in file{

```

```

chomp(lineSentence);
lineSentence <- "<p>\n<seg id=\"counter\">".lineSentence."</seg>\n</p>\n";

print to file_out (lineSentence);
counter++;
}

print to file_out ("</DOC>\n</refset>");
}

```

Gambar 4.18 Pseudocode Mengubah File Menjadi Reference File

Setiap berkas disimpan dalam format SGM dengan simbol pembungkus *srcset*, *tstset*, atau *refset*, tergantung pada jenis berkasnya. *Srcset* digunakan untuk *source file*, *tstset* digunakan untuk *test file*, dan *refset* digunakan untuk *reference file*. Setiap jenis berkas *sgm* memiliki parameter tertentu di dalam pembungkusnya masing-masing yang memberikan informasi tentang nama data set, bahasa sumber, dan bahasa tujuan. Informasi tentang nama data set disimpan pada simbol *setid*. Informasi tentang bahasa sumber disimpan pada simbol *srclang*. Informasi tentang bahasa tujuan disimpan pada *trglang*.

Program *Mteval-v11b.pl* dijalankan dengan cara mengetik "perl mteval-v11b.pl -r referensi.sgm -s source.sgm -t test.sgm > hasil evaluasi.txt". *Referensi.sgm* adalah *reference file*, *source.sgm* adalah *source file*, dan *test.sgm* adalah *test file*.

BAB 5 HASIL IMPLEMENTASI DAN UJI COBA

Bab ini menjelaskan tentang hasil implementasi dan uji coba yang dilakukan dalam penelitian. Pembahasan dimulai dari hasil dan analisis eksperimen arsitektur *baseline* (subbab 5.1), hasil dan analisis eksperimen arsitektur sistem POS *tagging* (subbab 5.2), hasil dan analisis eksperimen arsitektur sistem *parsing* (subbab 5.3), hasil dan analisis eksperimen arsitektur sistem *mixed* (subbab 5.4), rangkuman analisis eksperimen (subbab 5.5). Subbab pada bab ini penulis susun dalam urutan sistematis, sesuai dengan alur yang penulis lalui selama masa tugas akhir.

5.1 Hasil dan Analisis Eksperimen Arsitektur *Baseline*

Merujuk pada subbab 3.1.1 tentang arsitektur *baseline*, hasil eksperimen yang diperoleh dari arsitektur *baseline* dilaporkan pada Tabel 5.1 dan Tabel 5.2.

Pada Tabel 5.1 dan Tabel 5.2, "Percobaan 1" – "Percobaan 6" pada kolom pertama menandakan percobaan setiap fold pada korpus. Masing – masing korpus dibagi menjadi 6 buah fold (seperti penjelasan pada subbab 3.3). Nilai rata-rata BLEU dan NIST pada arsitektur *baseline* dijadikan sebagai standar acuan untuk perbandingan dengan hasil eksperimen arsitektur lainnya.

Tabel 5.1 Hasil Eksperimen Arsitektur *Baseline* Korpus Bible

Bible	BLEU	NIST
Percobaan 1	0.1377	4.7666
Percobaan 2	0.1540	4.6921
Percobaan 3	0.1877	5.3869
Percobaan 4	0.1578	5.1849
Percobaan 5	0.2222	5.9209
Percobaan 6	0.2394	6.174
Rata-Rata	0.1831	5.3542
Standard Deviasi	0.0407	0.6011

Tabel 5.1 merupakan tabel hasil evaluasi pada eksperimen arsitektur *baseline* dengan menggunakan korpus bible menurut penilaian BLEU dan NIST.

Tabel 5.2 Hasil Eksperimen Arsitektur *Baseline* Korpus Novel

Novel	BLEU	NIST
Percobaan 1	0.1827	5.2552
Percobaan 2	0.1754	5.2311
Percobaan 3	0.1793	5.3839
Percobaan 4	0.1927	5.4773
Percobaan 5	0.2032	5.6064
Percobaan 6	0.1895	5.3715
Rata-Rata	0.1871	5.3876
Standard Deviasi	0.0101	0.1402

Tabel 5.2 merupakan tabel hasil evaluasi pada eksperimen arsitektur *baseline* dengan menggunakan korpus novel menurut penilaian BLEU dan NIST.

5.2 Hasil dan Analisis Eksperimen Arsitektur Sistem POS *Tagging*

Merujuk pada subbab 3.1.2 tentang arsitektur sistem POS *tagging*, jumlah *word reordering* untuk korpus bible dan novel dilaporkan pada Tabel 5.3. Pada korpus bible terdapat 387.506 *token* kata dan pada korpus novel terdapat 147.953 *token* kata.

Tabel 5.3 Jumlah *Word Reordering* pada Korpus Bible dan Novel

Reordering	Total <i>Word Reordering</i>	
	Bible	Novel
JJx-NNx	6193	6993
NNx-NNx	1892	3227
PRP\$-NNx	12091	2544
JJx-NNx dan NNx-NNx	7969	9514
JJx-NNx dan PRP\$-NNx	18284	9537
NNx-NNx dan PRP\$-NNx	13651	5609
JJx-NNx, NNx-NNx, dan PRP\$-NNx	19728	11896

Tabel 5.3 merupakan tabel yang berisi data total jumlah *word reordering* untuk keenam buah fold pada masing – masing korpus.

Hasil eksperimen yang diperoleh dari arsitektur sistem POS *tagging* dilaporkan pada Tabel 5.4, Tabel 5.5, Tabel 5.6, dan Tabel 5.7.

Tabel 5.4 Hasil Eksperimen Arsitektur Sistem POS *Tagging* Korpus Bible dengan BLEU

Bible BLEU	JJx-NNx	NNx-NNx	PRP\$-NNx	JJx-NNx dan NNx-NNx	JJx-NNx dan PRP\$-NNx	NNx-NNx dan PRP\$-NNx	JJx-NNx , NNx-NNx dan PRP\$-NNx
Percobaan 1	0.1374	0.1353	0.1374	0.1366	0.1383	0.1351	0.1376
Percobaan 2	0.1589	0.155	0.1579	0.1614	0.1576	0.1543	0.1602
Percobaan 3	0.1883	0.1886	0.1878	0.1872	0.189	0.1885	0.1894
Percobaan 4	0.1574	0.1567	0.157	0.1584	0.1588	0.1574	0.1579
Percobaan 5	0.2219	0.2232	0.2225	0.2219	0.2213	0.2215	0.2215
Percobaan 6	0.2419	0.2409	0.2413	0.2423	0.2438	0.2429	0.2426
Rata-Rata	0.1843	0.1833	0.1840	0.1846	0.1848	0.1833	0.1849
Standard Deviasi	0.0408	0.0418	0.0409	0.0406	0.0410	0.0421	0.0407

Tabel 5.4 merupakan tabel hasil penilaian BLEU pada eksperimen arsitektur sistem POS *tagging* dengan menggunakan korpus bible.

Tabel 5.5 Hasil Eksperimen Arsitektur Sistem POS *Tagging* Korpus Bible dengan NIST

Bible BLEU	JJx-NNx	NNx-NNx	PRP\$-NNx	JJx-NNx dan NNx-NNx	JJx-NNx dan PRP\$-NNx	NNx-NNx dan PRP\$-NNx	JJx-NNx , NNx-NNx dan PRP\$-NNx
Percobaan 1	4.7826	4.7535	4.7595	4.7555	4.7707	4.7362	4.7599
Percobaan 2	4.7371	4.7331	4.7465	4.7594	4.757	4.7148	4.7634
Percobaan 3	5.3859	5.3691	5.38	5.3628	5.3766	5.3617	5.3756
Percobaan 4	5.1756	5.1686	5.1536	5.1742	5.1729	5.1723	5.1567
Percobaan 5	5.9089	5.911	5.8857	5.8981	5.8873	5.8938	5.8823
Percobaan 6	6.1831	6.1765	6.1379	6.1979	6.1793	6.1635	6.158
Rata-Rata	5.3622	5.3520	5.3439	5.3580	5.3573	5.3404	5.3493
Standard Deviasi	0.5890	0.5946	0.5762	0.5920	0.5823	0.5948	0.5773

Tabel 5.5 merupakan tabel hasil penilaian NIST pada eksperimen arsitektur sistem POS *tagging* dengan menggunakan korpus bible.

Tabel 5.6. Hasil Eksperimen Arsitektur Sistem POS *Tagging* Korpus Novel dengan BLEU

Bible BLEU	JJx-NNx	NNx-NNx	PRP\$-NNx	JJx-NNx dan NNx-NNx	JJx-NNx dan PRP\$-NNx	NNx-NNx dan PRP\$-NNx	JJx-NNx , NNx-NNx dan PRP\$-NNx
Percobaan 1	0.1855	0.1819	0.1838	0.1881	0.1869	0.185	0.1878
Percobaan 2	0.1807	0.178	0.181	0.1824	0.1789	0.179	0.1831
Percobaan 3	0.1813	0.1782	0.1797	0.1806	0.1829	0.1784	0.1828
Percobaan 4	0.1919	0.1874	0.1905	0.194	0.1911	0.1896	0.1928
Percobaan 5	0.2046	0.2008	0.2039	0.2042	0.205	0.204	0.2056
Percobaan 6	0.19	0.1889	0.191	0.1887	0.1902	0.189	0.1916
Rata-Rata	0.1890	0.1859	0.1883	0.1897	0.1892	0.1875	0.1906
Standard Deviasi	0.0089	0.0086	0.0090	0.0086	0.0090	0.0094	0.0084

Tabel 5.6 merupakan tabel hasil penilaian BLEU pada eksperimen arsitektur sistem POS *tagging* dengan menggunakan korpus novel.

Tabel 5.7 Hasil Eksperimen Arsitektur Sistem POS *Tagging* Korpus Novel dengan NIST

Bible BLEU	JJx-NNx	NNx-NNx	PRP\$-NNx	JJx-NNx dan NNx-NNx	JJx-NNx dan PRP\$-NNx	NNx-NNx dan PRP\$-NNx	JJx-NNx , NNx-NNx dan PRP\$-NNx
Percobaan 1	5.2721	5.2384	5.2719	5.3259	5.2975	5.2844	5.3097
Percobaan 2	5.2821	5.2628	5.3057	5.2704	5.2639	5.2643	5.2888
Percobaan 3	5.4164	5.3871	5.3816	5.4073	5.436	5.3905	5.442
Percobaan 4	5.4718	5.4233	5.4611	5.5157	5.4793	5.4571	5.4941
Percobaan 5	5.6393	5.6197	5.6279	5.6505	5.6353	5.6446	5.6414
Percobaan 6	5.3828	5.3795	5.3796	5.3565	5.3878	5.3845	5.4139
Rata-Rata	5.4108	5.3851	5.4046	5.4211	5.4166	5.4042	5.4317
Standard Deviasi	0.1361	0.1364	0.1277	0.1398	0.1345	0.1379	0.1293

Tabel 5.7 merupakan tabel hasil penilaian NIST pada eksperimen arsitektur sistem POS *tagging* dengan menggunakan korpus novel.

Tabel 5.8 Perbandingan *Word Reordering* Pada Korpus Bible

No	Bible	BLEU		NIST	
		Nilai Rata - Rata	Perubahan (%)	Nilai Rata - Rata	Perubahan (%)
1	<i>Baseline</i>	0.1831		5.3542	
2	JJx-NNx	0.1843	+0.6554%	5.3622	+0.1494%
3	NNx-NNx	0.1833	+0.1092%	5.3520	-0.0411%
4	PRP\$-NNx	0.1840	+0.4915%	5.3439	-0.1924%
5	JJx-NNx dan NNx-NNx	0.1846	+0.8192%	5.3580	+0.0710%
6	JJx-NNx dan PRP\$-NNx	0.1848	+0.9285%	5.3573	+0.0579%
7	NNx-NNx dan PRP\$-NNx	0.1833	+0.1092%	5.3404	-0.2577%
8	JJx-NNx, NNx-NNx dan PRP\$-NNx	0.1849	+0.9831%	5.3493	-0.0915%

Tabel 5.8 merupakan tabel perbandingan antara *word reordering* dengan *baseline* pada korpus bible. Tanda (+) pada kotak yang diarsir hijau menandakan peningkatan dan tanda (-) pada kotak yang diarsir biru menandakan penurunan.

Berdasarkan Tabel 5.8, peningkatan akurasi nilai BLEU pada korpus bible terdapat pada semua percobaan (percobaan 2-8). Peningkatan nilai BLEU terbesar terdapat pada percobaan kombinasi JJx-NNx, NNx-NNx dan PRP\$-NNx (percobaan 8).

$$\begin{aligned}
 \text{Persentase peningkatan terbesar yang terjadi} &= ((0.1849-0.1831) / 0.1831) * 100\% \\
 &= (0.0018 / 0.1831) * 100\% \\
 &= +0.9831\%
 \end{aligned}$$

Berdasarkan Tabel 5.8, peningkatan akurasi nilai NIST pada korpus bible, terdapat pada percobaan JJx-NNx (percobaan 2), JJx-NNx dan NNx-NNx (percobaan 5), JJx-NNx dan PRP\$-NNx (percobaan 6). Peningkatan nilai NIST terbesar terdapat pada percobaan JJx-NNx (percobaan 2).

$$\begin{aligned}
 \text{Persentase peningkatan terbesar yang terjadi} &= ((5.3622-5.3542) / 5.3542) * 100\% \\
 &= (0.008 / 5.3542) * 100\% \\
 &= +0.1494\%
 \end{aligned}$$

Tabel 5.8 direpresentasikan dalam bentuk grafik pada Gambar 5.1.

Tabel 5.9. Perbandingan *Word Reordering* Pada Korpus Novel

No	Novel	BLEU		NIST	
		Nilai Rata - Rata	Perubahan (%)	Nilai Rata - Rata	Perubahan (%)
1	<i>Baseline</i>	0.1871		5.3876	
2	JJx-NNx	0.1890	+1.0155%	5.4108	+0.4306%
3	NNx-NNx	0.1859	-0.6414%	5.3851	-0.0464%
4	PRP\$-NNx	0.1883	+0.6414%	5.4046	+0.3155%
5	JJx-NNx dan NNx-NNx	0.1897	+1.3896%	5.4211	+0.6218%
6	JJx-NNx dan PRP\$-NNx	0.1892	+1.1224%	5.4166	+0.5383%
7	NNx-NNx dan PRP\$-NNx	0.1875	+0.2138%	5.4042	+0.3081%
8	JJx-NNx, NNxNNx dan PRP\$-NNx	0.1906	+1.8707%	5.4317	+0.8185%

Tabel 5.9 merupakan tabel perbandingan antara *word reordering* dengan *baseline* pada korpus novel. Tanda (+) pada kotak yang diarsir hijau menandakan peningkatan dan tanda (-) pada kotak yang diarsir biru menandakan penurunan.

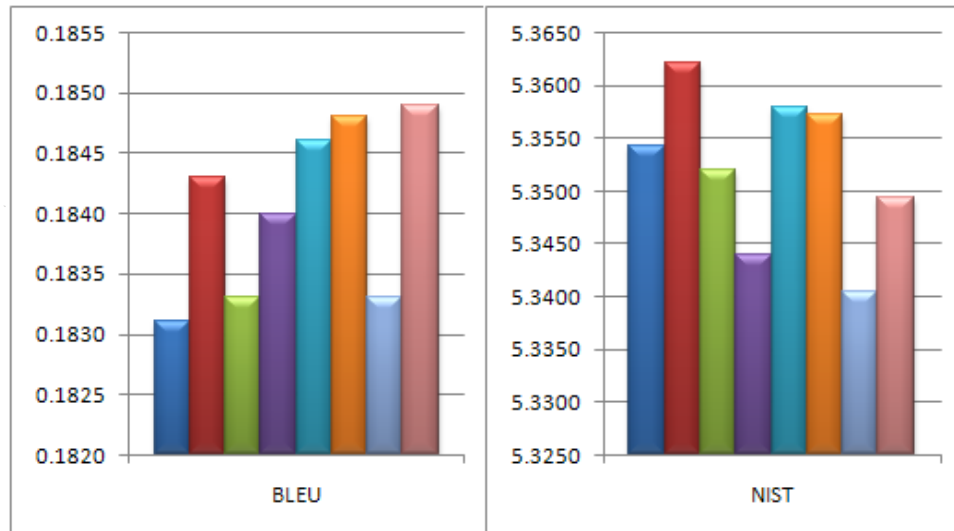
Berdasarkan Tabel 5.9, peningkatan akurasi nilai BLEU pada korpus novel terdapat pada semua percobaan kecuali NNx-NNx (percobaan 2, 4, 5, 6, 7, 8). Peningkatan nilai BLEU terbesar terdapat pada percobaan kombinasi JJx-NNx, NNxNNx dan PRP\$-NNx (percobaan 8).

$$\begin{aligned}
 \text{Persentase peningkatan terbesar yang terjadi} &= ((0.1906-0.1871) / 0.1871) * 100\% \\
 &= (0.0035 / 0.1871) * 100\% \\
 &= +1.8707\%
 \end{aligned}$$

Berdasarkan Tabel 5.9, peningkatan akurasi nilai NIST pada korpus novel terdapat pada semua percobaan kecuali NNx-NNx (percobaan 2, 4, 5, 6, 7, 8). Peningkatan nilai NIST terbesar terdapat pada percobaan kombinasi JJx-NNx, NNxNNx dan PRP\$-NNx (percobaan 8).

$$\begin{aligned}
 \text{Persentase peningkatan terbesar yang terjadi} &= ((5.4317-5.3876) / 5.3876) * 100\% \\
 &= (0.0441 / 5.3876) * 100\% \\
 &= +0.8185\%
 \end{aligned}$$

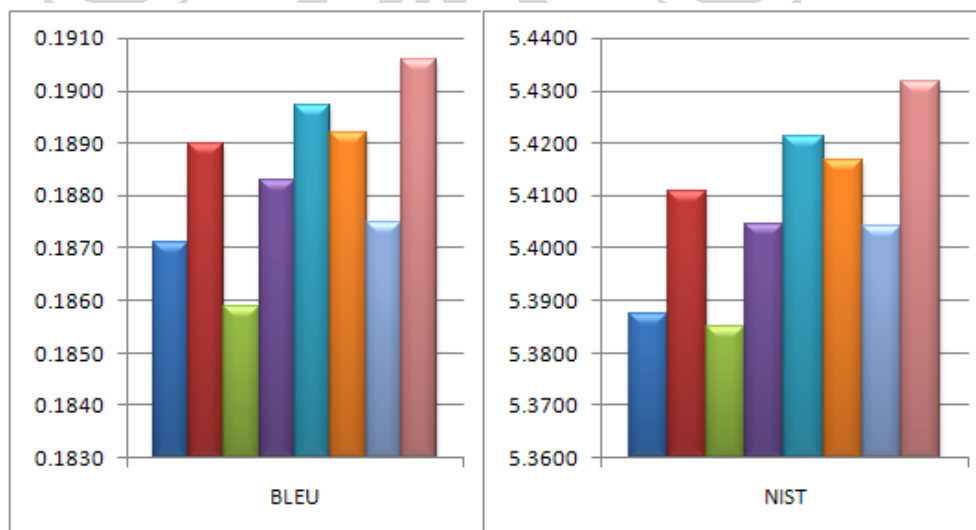
Tabel 5.9 direpresentasikan dalam bentuk grafik pada Gambar 5.2.



Gambar 5.1 Grafik Perbandingan Hasil Eksperimen *Word Reordering* Pada Korpus Bible

Keterangan:

■ Baseline	■ JJx-NNx
■ NNx-NNx	■ PRPŞ-NNx
■ JJx-NNx dan NNx-NNx	■ JJx-NNx dan PRPŞ-NNx
■ NNx-NNx dan PRPŞ-NNx	■ JJx-NNx, NNx-NNx dan PRPŞ-NNx



Gambar 5.2 Grafik Perbandingan Hasil Eksperimen *Word Reordering* Pada Korpus Novel

Keterangan:

■ Baseline	■ JJx-NNx
■ NNx-NNx	■ PRPŞ-NNx
■ JJx-NNx dan NNx-NNx	■ JJx-NNx dan PRPŞ-NNx
■ NNx-NNx dan PRPŞ-NNx	■ JJx-NNx, NNx-NNx dan PRPŞ-NNx

Dari Gambar 5.1, rancangan aturan *word reordering* yang dapat meningkatkan akurasi dan kualitas penerjemahan mesin penerjemah statistik pada korpus bible berdasarkan BLEU dan NIST, adalah:

1. Penukaran kata sifat dengan kata benda (JJx-NNx)
2. Penukaran kata sifat dengan kata benda dan kata benda dengan kata benda (JJx-NNx dan NNx-NNx)
3. Penukaran kata sifat dengan kata benda dan kata kepunyaan dengan kata benda (JJx-NNx dan PRP\$-NNx)

Dari Gambar 5.2, rancangan aturan *word reordering* yang dapat meningkatkan akurasi dan kualitas penerjemahan mesin penerjemah statistik pada korpus novel berdasarkan BLEU dan NIST, adalah:

1. Penukaran kata sifat dengan kata benda (JJx-NNx)
2. Penukaran kata kepunyaan dengan kata benda (PRP\$-NNx)
3. Penukaran kata sifat dengan kata benda dan kata benda dengan kata benda (JJx-NNx dan NNx-NNx)
4. Penukaran kata sifat dengan kata benda dan kata kepunyaan dengan kata benda (JJx-NNx dan PRP\$-NNx)
5. Penukaran kata benda dengan kata benda dan kata kepunyaan dengan kata benda (NNx-NNx dan PRP\$-NNx)
6. Penukaran kata sifat dengan kata benda, kata benda dengan kata benda dan kata kepunyaan dengan kata benda (JJx-NNx, NNx-NNx dan PRP\$-NNx)

Dari keseluruhan hasil eksperimen percobaan *word reordering* pada kedua buah korpus dapat dilihat terdapat beberapa aturan *word reordering* yang dapat meningkatkan kualitas dan akurasi dari mesin penerjemah statistik. Rancangan aturan *word reordering* tersebut adalah:

1. Penukaran kata sifat dengan kata benda (JJx-NNx)
2. Penukaran kata sifat dengan kata benda dan kata benda dengan kata benda (JJx-NNx dan NNx-NNx)

3. Penukaran kata sifat dengan kata benda dan kata kepunyaan dengan kata benda (JJx-NNx dan PRP\$-NNx)

Tabel 5.10 Rancangan Aturan Word Reordering yang Memberikan Peningkatan

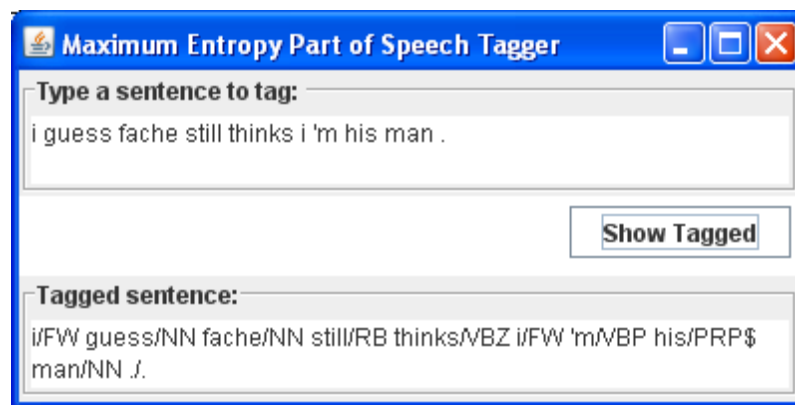
Bible	BLEU		NIST	
	Nilai Rata - Rata	Perubahan (%)	Nilai Rata - Rata	Perubahan (%)
<i>Baseline</i>	0.1831		5.3542	
JJx-NNx	0.1843	+0.6554%	5.3622	+0.1494%
JJx-NNx dan NNx-NNx	0.1846	+0.8192%	5.3580	+0.0710%
JJx-NNx dan PRP\$-NNx	0.1848	+0.9285%	5.3573	+0.0579%
Novel	BLEU		NIST	
	Nilai Rata - Rata	Perubahan (%)	Nilai Rata - Rata	Perubahan (%)
<i>Baseline</i>	0.1871		5.3876	
JJx-NNx	0.1890	+1.0155%	5.4108	+0.4306%
JJx-NNx dan NNx-NNx	0.1897	+1.3896%	5.4211	+0.6218%
JJx-NNx dan PRP\$-NNx	0.1892	+1.1224%	5.4166	+0.5383%

Dari ketiga buah rancangan aturan *word reordering* yang memberikan peningkatan pada kedua buah korpus baik menurut BLEU dan NIST, ketiga buah aturan tersebut mengandung penukaran kata sifat dengan kata benda (JJx-NNx). Rancangan aturan dasar JJx-NNx tersebut memberikan kontribusi peningkatan pada kualitas dan akurasi dari mesin penerjemah statistik.

5.2.1 Analisis Kesalahan Eksperimen Arsitektur Sistem POS *Tagging*

Ada beberapa faktor yang menyebabkan terjadinya *error* dan penurunan kualitas dan akurasi pada arsitektur sistem POS *tagging* dan rancangan *word reordering*, yaitu:

- 1 Kesalahan *tagging* dari Stanford POS Tagger. Berikut adalah contoh kesalahan *tagging* untuk kata "guess" pada kalimat " i guess fache still thinks i'm his man ." yang diambil pada segment 10004 dari korpus novel.



Gambar 5.3 Contoh Kesalahan Tagging Stanford POS Tagger

Kata "guess" seharusnya diberi tag /VBP yang berarti *verb present* atau kata kerja bukan diberi tag /NN yang berarti *common noun* atau kata benda. Keterbatasan Stanford POS Tagging ini dapat menyebabkan *error* pada proses *word reordering*.

- 2 Beberapa kata Bahasa Inggris pada rujukan yang seharusnya diterjemahkan menjadi Bahasa Indonesia, tidak diterjemahkan menjadi Bahasa Indonesia. Kata tersebut tetap ditulis dalam Bahasa Inggris. Berikut adalah contoh penurunan *score* pada segment 127 dari korpus novel. Sebelum dilakukan JJx-NNx *reordering* nilai BLEU adalah 0.7071. Setelah dilakukan NNx-NNx *reordering* nilai BLEU menurun menjadi 0.3156.

S	we found your name in his daily planner .
SR	we found your name in his planner daily .
T	kami menemukan nama anda dalam daily planner .
TR	kami menemukan nama anda di atas planner harian .
R	kami menemukan nama anda dalam daily planner-nya .

S adalah *source* yang merupakan kalimat Bahasa Inggris. SR adalah *source reordered* yang merupakan kalimat Bahasa Inggris dengan proses *reordering*. T adalah *test* yang merupakan kalimat terjemahan Bahasa Indonesia dari mesin penerjemah statistik tanpa proses *reordering*. TR adalah *test reordered* yang merupakan kalimat terjemahan Bahasa Indonesia dari mesin penerjemah statistik dengan proses *reordering*. R adalah *reference* yang merupakan kalimat rujukan Bahasa Indonesia. Pada contoh di atas kata "daily planner" diterjemahkan juga menjadi daily planner dalam Bahasa Indonesia.

- 3 Proses penukaran nama orang yang terdiri dari 2 kata yang berurutan dapat menyebabkan penurunan kualitas dan akurasi dari mesin penerjemah statistik. Hal tersebut dikarenakan 2 kata tersebut diberi *tag* /NN dan ditukar. Berikut adalah contoh penurunan *score* pada segment 1583 dari korpus novel. Sebelum dilakukan NNx-NNx *reordering* nilai BLEU adalah 0.2390. Setelah dilakukan NNx-NNx *reordering* nilai BLEU menurun menjadi 0.2310.

S	robert langdon hung up the phone , he looked ill.
SR	langdon robert hung up the phone , he looked ill .
T	robert langdon menutup teleponnya , dia melihat ill .
TR	langdon robert menutup teleponnya , dia melihat ill.
R	robert langdon menutup teleponnya .

- 4 Adanya kata – kata dengan pola yang seharusnya ditukar pada level frase mengalami penukaran pada proses word reordering ini. Berikut adalah contoh penurunan *score* pada segment 338 dari korpus novel. Sebelum dilakukan JJ-NNx *reordering* nilai BLEU adalah 0.2709. Setelah dilakukan JJ-NNx *reordering* nilai BLEU menurun menjadi 0.2701.

S	after all , he had somehow obtained very secret information .
SR	after all , he had somehow obtained very information secret .
T	lagi pula , dia sudah entah bagaimana obtained sangat rahasia informasi.
TR	lagi pula , dia telah bagiku obtained sangat informasi rahasia
R	lagi pula , dia telah mendapatkan informasi yang sangat rahasia .

Kata "very secret information" apabila dilakukan proses *tagging* maka memiliki kelas kata "very/RB secret/JJ information/NN". Pada proses *word reordering* akan ditukar menjadi "very/RB information/NN secret/JJ". Seharusnya kata "very secret information" dilakukan penukaran ADJP – NNx pada level frase menjadi "information very secret " yang diterjemahkan menjadi informasi yang sangat rahasia. Penukaran level frase inilah yang ditangani oleh *phrase reordering* pada arsitektur sistem *parsing*

5.3 Hasil dan Analisis Eksperimen Arsitektur Sistem *Parsing*

Merujuk pada subbab 3.1.3 tentang arsitektur sistem *parsing*, jumlah *phrase reordering* pada korpus bible dan novel dilaporkan pada Tabel 5.11.

Tabel 5.11 Jumlah *Phrase Reordering* pada Korpus Bible dan Novel

Reordering	Total <i>Phrase Reordering</i>	
	Bible	Novel
ADJP-NNx	172	345
NP-NP	9605	1695
ADJP-NP	134	53
ADJP-NNx dan NP-NP	9768	2028
ADJP-NNx dan ADJP-NP	312	399
ADJP-NP dan NP-NP	9736	1746
ADJP-NNx, NP-NP, dan ADJP-NP	9914	2085

Tabel 5.11 merupakan tabel yang berisi data jumlah *phrase reordering* untuk keenam buah fold pada masing – masing korpus.

Hasil eksperimen yang diperoleh dari arsitektur sistem *parsing* dilaporkan pada Tabel 5.12, Tabel 5.13, Tabel 5.14, dan Tabel 5.15.

Tabel 5.12 Hasil Eksperimen Arsitektur Sistem Parsing Korpus Bible dengan BLEU

Bible BLEU	ADJP - NN	NP - NP	ADJP - NP	ADJP - NN dan NP - NP	ADJP - NN dan ADJP - NP	ADJP - NP dan NP - NP	ADJP - NN, NP - NP, dan ADJP - NP
Percobaan 1	0.1355	0.1237	0.1365	0.1236	0.1344	0.1237	0.1241
Percobaan 2	0.1541	0.1369	0.1563	0.1375	0.1528	0.1413	0.1387
Percobaan 3	0.1877	0.1682	0.1878	0.1682	0.1893	0.1677	0.17
Percobaan 4	0.158	0.1423	0.1563	0.1402	0.1571	0.1414	0.1418
Percobaan 5	0.2214	0.2029	0.2212	0.2025	0.2211	0.201	0.2002
Percobaan 6	0.2426	0.2157	0.2431	0.2119	0.2439	0.2133	0.2124
Rata-Rata	0.1832	0.1650	0.1835	0.1640	0.1831	0.1647	0.1645
Standard Deviasi	0.0419	0.0375	0.0417	0.0366	0.0428	0.0359	0.0358

Tabel 5.12 merupakan tabel hasil penilaian BLEU pada eksperimen arsitektur sistem *parsing* dengan menggunakan korpus bible.

Tabel 5.13 Hasil Eksperimen Arsitektur Sistem *Parsing* Korpus Bible dengan NIST

Bible NIST	ADJP - NN	NP - NP	ADJP - NP	ADJP - NN dan NP - NP	ADJP - NN dan ADJP - NP	ADJP - NP dan NP - NP	ADJP - NN, NP - NP, dan ADJP - NP
Percobaan 1	4.7727	4.6003	4.7653	4.601	4.7393	4.5897	4.6059
Percobaan 2	4.6891	4.4778	4.7165	4.4897	4.6798	4.5165	4.5223
Percobaan 3	5.3942	5.1986	5.4025	5.179	5.3977	5.1837	5.2044
Percobaan 4	5.1739	5.0298	5.1611	4.9965	5.1757	4.9833	5.02
Percobaan 5	5.9176	5.6964	5.8917	5.6842	5.8954	5.6703	5.6555
Percobaan 6	6.2113	5.8654	6.2014	5.8211	6.2204	5.8189	5.8219
Rata-Rata	5.3598	5.1447	5.3564	5.1286	5.3514	5.1271	5.1383
Standard Deviasi	0.6110	0.5623	0.6000	0.5468	0.6180	0.5402	0.5322

Tabel 5.13 merupakan tabel hasil penilaian NIST pada eksperimen arsitektur sistem *parsing* dengan menggunakan korpus bible.

Tabel 5.14 Hasil Eksperimen Arsitektur Sistem *Parsing* Korpus Novel dengan BLEU

Novel BLEU	ADJP - NN	NP - NP	ADJP - NP	ADJP - NN dan NP - NP	ADJP - NN dan ADJP - NP	ADJP - NP dan NP - NP	ADJP - NN, NP - NP, dan ADJP - NP
Percobaan 1	0.1844	0.1755	0.1826	0.1741	0.1849	0.1739	0.1738
Percobaan 2	0.1756	0.1656	0.1767	0.1677	0.1778	0.1666	0.1666
Percobaan 3	0.179	0.1702	0.1799	0.171	0.1793	0.1708	0.1726
Percobaan 4	0.1929	0.1785	0.1917	0.178	0.1912	0.1803	0.1775
Percobaan 5	0.2028	0.1909	0.2033	0.1906	0.2026	0.1924	0.1899
Percobaan 6	0.1889	0.1799	0.1887	0.1797	0.1895	0.1804	0.1802
Rata-Rata	0.1873	0.1768	0.1872	0.1769	0.1876	0.1774	0.1768
Standard Deviasi	0.0099	0.0087	0.0097	0.0081	0.0091	0.0091	0.0079

Tabel 5.14 merupakan tabel hasil penilaian BLEU pada eksperimen arsitektur sistem *parsing* dengan menggunakan korpus novel.

Tabel 5.15 Hasil Eksperimen Arsitektur Sistem *Parsing* Korpus Novel dengan NIST

Novel NIST	ADJP - NN	NP - NP	ADJP - NP	ADJP - NN dan NP - NP	ADJP - NN dan ADJP - NP	ADJP - NP dan NP - NP	ADJP - NN, NP - NP, dan ADJP - NP
Percobaan 1	5.2456	5.1826	5.2315	5.1756	5.2449	5.1729	5.1575
Percobaan 2	5.2275	5.1671	5.2305	5.1834	5.2506	5.1895	5.1692
Percobaan 3	5.3551	5.3091	5.3701	5.333	5.3649	5.3107	5.3387
Percobaan 4	5.4613	5.3397	5.4568	5.345	5.4382	5.3524	5.3298
Percobaan 5	5.5915	5.5342	5.6057	5.5109	5.5985	5.5319	5.5079
Percobaan 6	5.3638	5.3173	5.3499	5.3068	5.3682	5.3166	5.313
Rata-Rata	5.3741	5.3083	5.3741	5.3091	5.3776	5.3123	5.3027
Standard Deviasi	0.1366	0.1325	0.1429	0.1235	0.1315	0.1299	0.1290

Tabel 5.15 merupakan tabel hasil penilaian NIST pada eksperimen arsitektur sistem *parsing* dengan menggunakan korpus novel.

Tabel 5.16 Perbandingan *Phrase Reordering* Pada Korpus Bible

No	Bible	BLEU		NIST	
		Nilai Rata - Rata	Perubahan (%)	Nilai Rata - Rata	Perubahan (%)
1	<i>Baseline</i>	0.1831		5.3542	
2	ADJP - NNx	0.1832	+0.0546%	5.3598	+0.1046%
3	NP - NP	0.1650	-9.8853%	5.1447	-3.9128%
4	ADJP - NP	0.1835	+0.2185%	5.3564	+0.0411%
5	ADJP - NNx dan NP - NP	0.1640	-10.4315%	5.1286	-4.2135%
6	ADJP - NNx dan ADJP - NP	0.1831	0.0000%	5.3514	-0.0523%
7	ADJP - NP dan NP - NP	0.1647	-10.0492%	5.1271	-4.2415%
8	ADJP - NNx, NP - NP, dan ADJP - NP	0.1645	-10.1584%	5.1383	-4.0323%

Tabel 5.16 merupakan tabel perbandingan antara *phrase reordering* dengan *baseline* pada korpus bible. Tanda (+) pada kotak yang diarsir hijau menandakan peningkatan dan tanda (-) pada kotak yang diarsir biru menandakan penurunan.

Berdasarkan Tabel 5.16, peningkatan akurasi nilai BLEU pada korpus bible terdapat pada percobaan ADJP-NNx (percobaan 2), ADJP-NP (percobaan 4). Peningkatan nilai BLEU terbesar terdapat pada percobaan ADJP-NP (percobaan 4).

$$\begin{aligned}
 \text{Persentase peningkatan terbesar yang terjadi} &= ((0.1835-0.1831) / 0.1831) * 100\% \\
 &= (0.0004 / 0.1831) * 100\% \\
 &= +0.2185\%
 \end{aligned}$$

Berdasarkan Tabel 5.16, peningkatan akurasi nilai NIST pada korpus bible terdapat pada percobaan ADJP-NNx (percobaan 2), ADJP-NP (percobaan 4). Peningkatan nilai BLEU terbesar terdapat pada percobaan ADJP-NNx (percobaan 2).

$$\begin{aligned}
 \text{Persentase peningkatan terbesar yang terjadi} &= ((5.3598-5.3542) / 5.3542) * 100\% \\
 &= (0.0056 / 5.3542) * 100\% \\
 &= +0.1046\%
 \end{aligned}$$

Tabel 5.16 direpresentasikan dalam bentuk grafik pada Gambar 5.4.

Tabel 5.17 Perbandingan *Phrase Reordering* Pada Korpus Novel

No	Novel	BLEU		NIST	
		Nilai Rata - Rata	Perubahan (%)	Nilai Rata - Rata	Perubahan (%)
1	<i>Baseline</i>	0.1871		5.3876	
2	ADJP - NNx	0.1873	+0.1069%	5.3741	-0.2506%
3	NP - NP	0.1768	-5.5051%	5.3083	-1.4719%
4	ADJP - NP	0.1872	+0.0534%	5.3741	-0.2506%
5	ADJP - NNx dan NP - NP	0.1769	-5.4516%	5.3091	-1.4570%
6	ADJP - NNx dan ADJP - NP	0.1876	+0.2672%	5.3776	-0.1856%
7	ADJP - NP dan NP - NP	0.1774	-5.1844%	5.3123	-1.3977%
8	ADJP - NNx, NP - NP, dan ADJP - NP	0.1768	-5.5051%	5.3027	-1.5758%

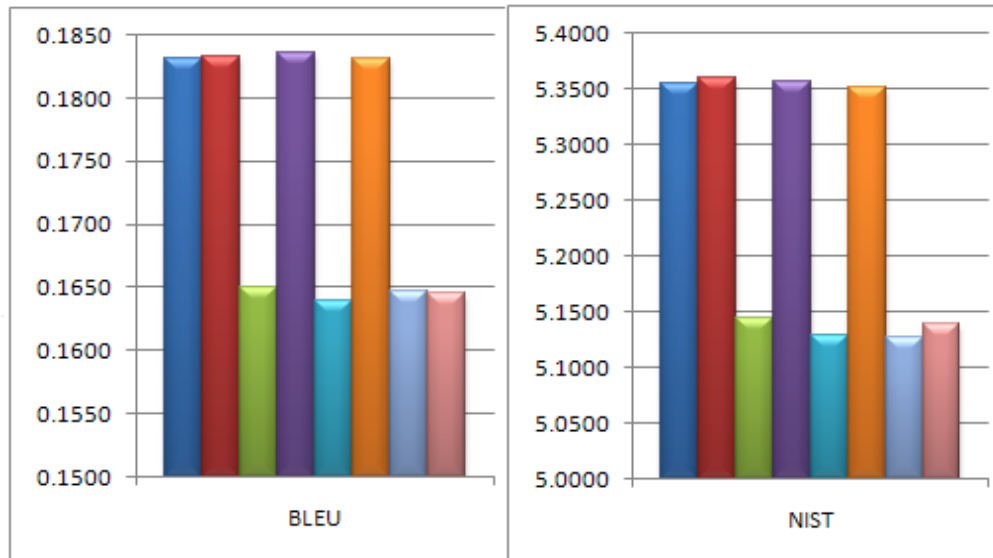
Tabel 5.17 merupakan tabel perbandingan antara *phrase reordering* dengan *baseline* pada korpus novel. Tanda (+) pada kotak yang diarsir hijau menandakan peningkatan dan tanda (-) pada kotak yang diarsir biru menandakan penurunan.

Berdasarkan Tabel 5.17, peningkatan akurasi nilai BLEU pada korpus novel terdapat pada percobaan ADJP-NNx (percobaan 2), ADJP-NP (percobaan 4), ADJP-NNx dan ADJP-NP (percobaan 6). Peningkatan nilai BLEU terbesar terdapat pada percobaan kombinasi ADJP-NNx dan ADJP-NP (percobaan 6).

$$\begin{aligned}
 \text{Persentase peningkatan terbesar yang terjadi} &= ((0.1876-0.1871) / 0.1871) * 100\% \\
 &= (0.0005 / 0.1871) * 100\% \\
 &= +0.2672\%
 \end{aligned}$$

Berdasarkan Tabel 5.17, tidak ada peningkatan akurasi nilai NIST pada korpus novel. Penurunan nilai NIST terkecil pada korpus novel terdapat pada percobaan ADJP-NNx dan ADJP - NP

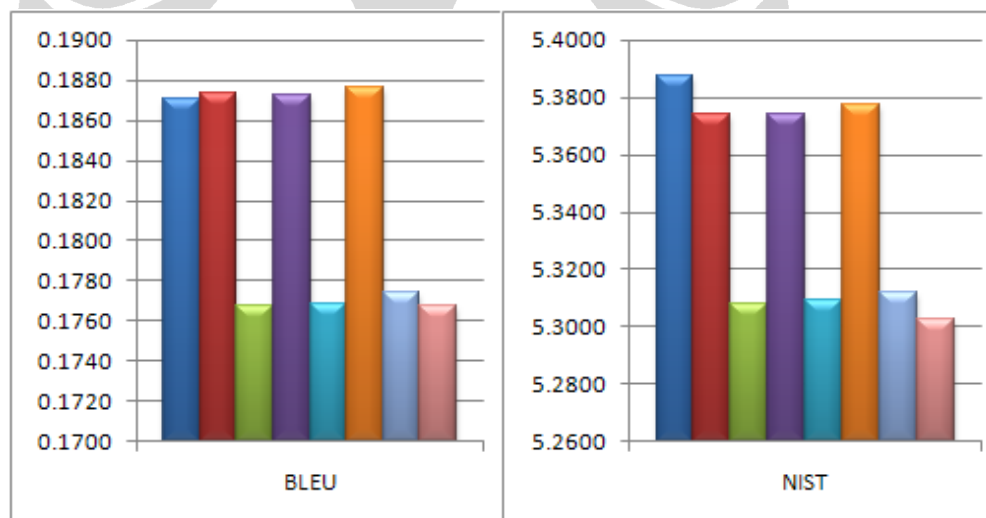
Tabel 5.17 direpresentasikan dalam bentuk grafik pada Gambar 5.5.



Gambar 5.4 Grafik Perbandingan Hasil Eksperimen *Phrase Reordering* Pada Korpus Bible

Keterangan:

- Baseline
- NP - NP
- ADJP - NNx dan NP - NP
- ADJP - NP dan NP - NP
- ADJP - NNx
- ADJP - NP
- ADJP - NNx dan ADJP - NP
- ADJP - NNx, NP - NP, dan ADJP - NP



Gambar 5.5 Grafik Perbandingan Hasil Eksperimen *Phrase Reordering* Pada Korpus Novel

Keterangan:

- Baseline
- NP - NP
- ADJP - NNx dan NP - NP
- ADJP - NP dan NP - NP
- ADJP - NNx
- ADJP - NP
- ADJP - NNx dan ADJP - NP
- ADJP - NNx, NP - NP, dan ADJP - NP

Dari Gambar 5.4, rancangan aturan *phrase reordering* yang dapat meningkatkan akurasi dan kualitas penerjemahan mesin penerjemah statistik pada korpus bible berdasarkan BLEU dan NIST, adalah:

1. Penukaran frase sifat dengan kata benda (ADJP-NN_x)
2. Penukaran frase sifat dengan frase benda (ADJP-NP)

Dari Gambar 5.5, rancangan aturan *phrase reordering* yang dapat meningkatkan akurasi dan kualitas penerjemahan mesin penerjemah statistik pada korpus novel berdasarkan BLEU, adalah:

1. Penukaran frase sifat dengan kata benda (ADJP-NN_x)
2. Penukaran frase sifat dengan frase benda (ADJP-NP)
3. Penukaran frase sifat dengan kata benda dan frase sifat dengan frase benda (ADJP-NN_x dan ADJP-NP)

Dari Gambar 5.5, tidak ada rancangan aturan *phrase reordering* yang dapat meningkatkan akurasi dan kualitas penerjemahan mesin penerjemah statistik pada korpus novel berdasarkan NIST.

Dari keseluruhan hasil eksperimen percobaan *phrase reordering* pada kedua buah korpus dapat dilihat terdapat 3 buah rancangan aturan *phrase reordering* yang lebih baik daripada rancangan aturan *phrase reordering* lainnya. Rancangan aturan *phrase reordering* tersebut adalah:

1. Penukaran frase sifat dengan kata benda (ADJP-NN_x)
2. Penukaran frase sifat dengan frase benda (ADJP-NP)
3. Penukaran frase sifat dengan kata benda dan frase sifat dengan frase benda (ADJP-NN_x dan ADJP-NP)

Akan tetapi dari ketiga buah rancangan aturan *phrase reordering* tersebut tidak ada yang berhasil meningkatkan akurasi dan kualitas penerjemahan mesin penerjemah statistik (BLEU dan NIST) pada kedua buah korpus.

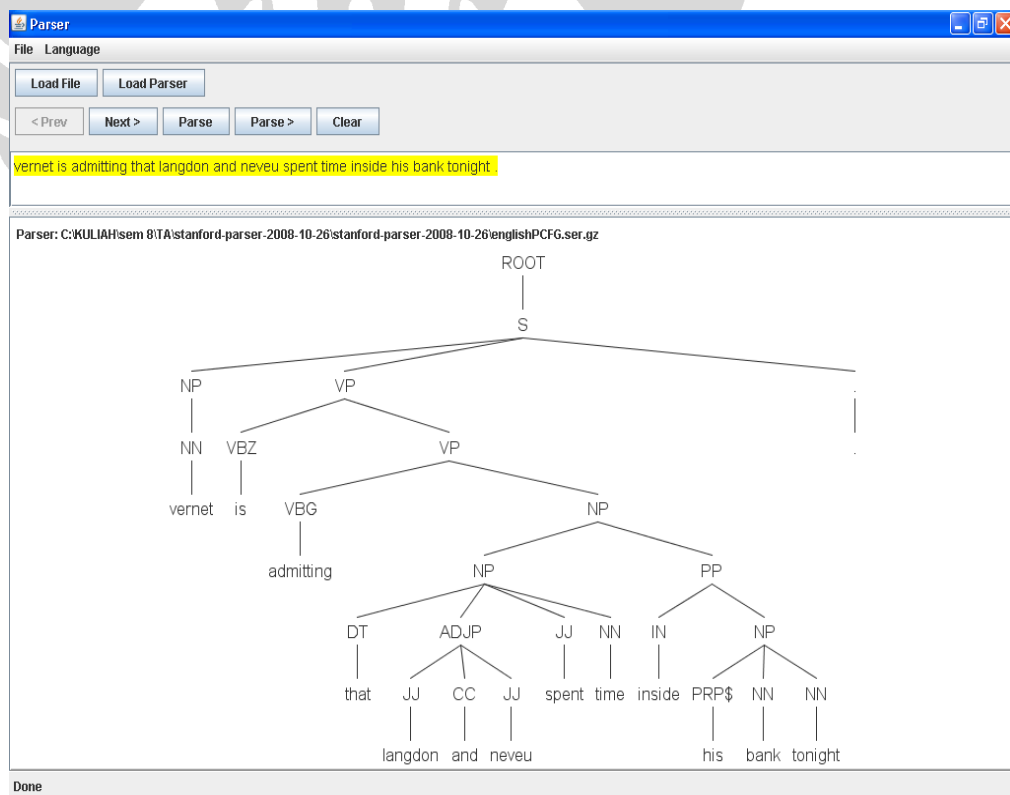
Dari keseluruhan hasil eksperimen tersebut dapat dilihat juga bahwa rancangan aturan dasar dan kombinasi *phrase reordering* NP-NP memberikan penurunan yang besar pada akurasi dan kualitas penerjemahan mesin penerjemah statistik.

Peningkatan akurasi dan kualitas penerjemahan yang diperoleh arsitektur sistem POS *tagging* dengan *word reordering* lebih besar dan lebih baik daripada arsitektur sistem *parsing* dengan *phrase reordering*.

5.3.1 Analisis Kesalahan Eksperimen Arsitektur Sistem *Parsing*

Ada beberapa faktor yang menyebabkan terjadinya *error* dan penurunan kualitas dan akurasi pada arsitektur sistem *parsing* dan rancangan *phrase reordering*, yaitu:

- 1 Kesalahan *parsing* struktur *tree* dari Stanford Parser. Berikut adalah contoh kesalahan *parsing* pada kalimat "vernet is admitting that langdon and neveu spent time inside his bank tonight."



Gambar 5.6 Contoh Kesalahan Parsing Stanford POS Tagger

Kelompok kata "langdon and neveu" diberi simbol sebagai frase sifat atau ADJP. Seharusnya kelompok kata tersebut diberi simbol sebagai frase benda atau NP. Hal ini dapat menyebabkan penurunan akurasi dan kualitas dari mesin penerjemah statistik.

2. Proses *phrase reordering* memiliki kelemahan yang menyebabkan penurunan akurasi dan kualitas penerjemahan. Penurunan akurasi dan kualitas penerjemahan tersebut terjadi pada penukaran pada pasangan frase yang masih memiliki *parent* yang sama dengan jarak yang jauh. Berikut adalah contoh penurunan *score* pada segment 2470 novel. Sebelum dilakukan ADJP-NP dan NP-NP *reordering* nilai BLEU adalah 0.4131. Setelah dilakukan ADJP-NP dan NP-NP *reordering* nilai BLEU menurun menjadi 0.2217.

S	the symbolism of the clues meshed <u>too perfectly</u> , the pentacle , <u>the vitruvian man</u> , da vinci , <u>the goddess</u> , <u>and even the fibonacci sequence</u> .
SR	the symbolism of the clues meshed <u>the vitruvian man</u> , the pentacle , <u>the goddess</u> , da vinci , <u>and even the fibonacci sequence</u> , <u>too perfectly</u> .
T	simbolisme yang meshed terlalu sempurna , pentakel , vitruvian man itu , da vinci , dewi , dan bahkan fibonacci .
TR	simbolisme petunjuk meshed vitruvian man itu , pentakel , dewi , da vinci , dan bahkan fibonacci , terlalu sempurna .
R	simbolisme dari petunjuk-petunjuk itu bertautan dengan sangat sempurna , pentakel ,the vitruvian man , da vinci , dewi , dan bahkan deret ukur fibonacci .

Kata "too perfectly" pada kalimat di atas mengalami penukaran posisi. Perbedaan letak kata "terlalu sempurna" pada hasil terjemahan dengan kata rujukan "sangat sempurna" yang cukup jauh dapat menyebabkan penurunan nilai akurasi dan kualitas penerjemahan. Kelemahan inilah yang mengakibatkan kurang berhasilnya proses *phrase reordering* (Penukaran antar konstituen yang terlalu jauh).

5.4 Hasil dan Analisis Eksperimen Arsitektur Sistem Mixed

Merujuk pada subbab 3.1.4 tentang arsitektur *mixed*, jumlah *phrase reordering* dan *word reordering* yang terjadi pada korpus bible dan novel dilaporkan pada Tabel 5.18.

Tabel 5.18 Jumlah Phrase Reordering dan Word Reordering pada Korpus Bible dan Novel

Reordering	Bible		Novel	
	Phrase Reordering	Word Reordering	Phrase Reordering	Word Reordering
ADJP-NNx dan JJx-NNx	172	5878	345	6787
ADJP-NNx dan PRP\$-NNx	172	12116	345	2584

Tabel 5.18 merupakan tabel yang berisi data jumlah *phrase reordering* dan *word reordering* untuk keenam buah fold pada masing – masing korpus.

Hasil eksperimen yang diperoleh dari arsitektur sistem *mixed* dilaporkan pada Tabel 5.19, Tabel 5.20.

Tabel 5.19 Hasil Eksperimen Arsitektur Sistem Mixed Korpus Bible

Bible	BLEU		NIST	
	ADJP-NNx dan JJx-NNx	ADJP-NNx dan PRP\$-NNx	ADJP-NNx dan JJx-NNx	ADJP-NNx dan PRP\$-NNx
Percobaan 1	0.1353	0.1352	4.7414	4.7278
Percobaan 2	0.1596	0.1573	4.7296	4.6917
Percobaan 3	0.1871	0.1883	5.377	5.3896
Percobaan 4	0.1587	0.1592	5.1863	5.187
Percobaan 5	0.2228	0.2225	5.8901	5.8886
Percobaan 6	0.2421	0.2403	6.1613	6.1363
Rata-Rata	0.1843	0.1838	5.3476	5.3368
Standard Deviasi	0.0412	0.0409	0.5887	0.5926

Tabel 5.19 merupakan tabel hasil evaluasi pada eksperimen arsitektur *mixed* dengan menggunakan korpus bible menurut penilaian BLEU dan NIST.

Tabel 5.20 Hasil Eksperimen Arsitektur Sistem Mixed Korpus Novel

Novel	BLEU		NIST	
	ADJP-NNx dan JJx-NNx	ADJP-NNx dan PRP\$-NNx	ADJP-NNx dan JJx-NNx	ADJP-NNx dan PRP\$-NNx
Percobaan 1	0.1868	0.1843	5.2898	5.2898
Percobaan 2	0.1819	0.1790	5.2944	5.2515
Percobaan 3	0.1802	0.1804	5.4	5.3829
Percobaan 4	0.1935	0.1916	5.4788	5.4745
Percobaan 5	0.2045	0.2029	5.6301	5.6296
Percobaan 6	0.1879	0.1888	5.3552	5.3774
Rata-Rata	0.1891	0.1878	5.4081	5.4010
Standard Deviasi	0.0089	0.0088	0.1297	0.1366

Tabel 5.20 merupakan tabel hasil evaluasi pada eksperimen arsitektur *mixed* dengan menggunakan korpus novel menurut penilaian BLEU dan NIST.

Tabel 5.21 Perbandingan *Mixed* Pada Korpus Bible

No	Bible	BLEU		NIST	
		Nilai Rata - Rata	Perubahan (%)	Nilai Rata - Rata	Perubahan (%)
1	<i>Baseline</i>	0.1831		5.3542	
2	ADJP-NNx dan JJx-NNx	0.1843	+0.6554%	5.3476	-0.1233%
3	ADJP-NNx dan PRP\$-NNx	0.1838	+0.3823%	5.3368	-0.3250%

Tabel 5.21 merupakan tabel perbandingan antara *mixed* dengan *baseline* pada korpus bible. Tanda (+) pada kotak yang diarsir hijau menandakan peningkatan dan tanda (-) pada kotak yang diarsir biru menandakan penurunan.

Berdasarkan Tabel 5.21, peningkatan akurasi nilai BLEU pada korpus bible terdapat pada semua percobaan (percobaan 2 dan 3). Peningkatan nilai BLEU terbesar terdapat pada percobaan ADJP-NNx dan JJx-NNx (percobaan 2).

$$\begin{aligned}
 \text{Persentase peningkatan yang terjadi} &= ((0.1843 - 0.1831) / 0.1831) * 100\% \\
 &= (0.0012 / 0.1831) * 100\% \\
 &= +0.6554 \%
 \end{aligned}$$

Berdasarkan Tabel 5.21, tidak ada peningkatan akurasi nilai NIST pada korpus bible. Penurunan nilai NIST terkecil pada korpus bible terdapat pada percobaan ADJP-NNx dan JJx-NNx (percobaan 2).

$$\begin{aligned}
 \text{Persentase penurunan yang terjadi} &= ((5.3476 - 5.3542) / 5.3542) * 100\% \\
 &= (-0.0066 / 5.3542) * 100\% \\
 &= -0.1233 \%
 \end{aligned}$$

Perbandingan Tabel 5.21, ADJP-NNx, JJx-NNx, PRP\$-NNx direpresentasikan dalam bentuk grafik pada Gambar 5.7.

Tabel 5.22 Perbandingan *Mixed* Pada Korpus Novel

No	Novel	BLEU		NIST	
		Nilai Rata - Rata	Perubahan (%)	Nilai Rata - Rata	Perubahan (%)
1	<i>Baseline</i>	0.1871		5.3876	
2	ADJP-NNx dan JJx-NNx	0.1891	+1.0689%	5.4081	+0.3796%
3	ADJP-NNx dan PRP\$-NNx	0.1878	+0.3741%	5.401	+0.2487%

Tabel 5.22 merupakan tabel perbandingan antara *phrase and word reordering* dengan *baseline* pada korpus novel. Tanda (+) pada kotak yang diarsir hijau menandakan peningkatan dan tanda (-) pada kotak yang diarsir biru menandakan penurunan.

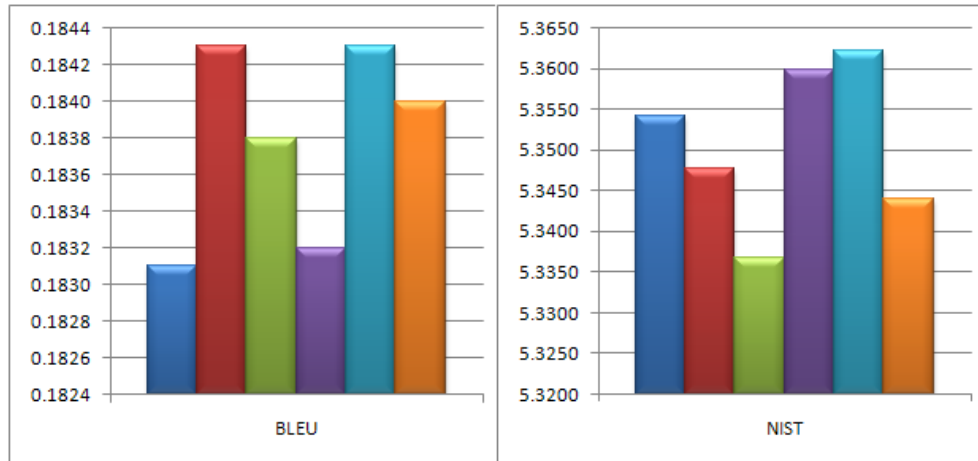
Berdasarkan Tabel 5.22, peningkatan akurasi nilai BLEU pada korpus bible terdapat pada semua percobaan (percobaan 2 dan 3). Peningkatan akurasi nilai BLEU terbesar pada korpus novel terdapat percobaan ADJP-NNx dan JJx-NNx (percobaan 2).

$$\begin{aligned}
 \text{Persentase peningkatan yang terjadi} &= ((0.1891 - 0.1871) / 0.1871) * 100\% \\
 &= (0.002 / 0.1871) * 100\% \\
 &= +1.0689 \%
 \end{aligned}$$

Berdasarkan Tabel 5.22, peningkatan akurasi nilai NIST pada korpus bible terdapat pada semua percobaan (percobaan 2 dan 3). peningkatan akurasi nilai NIST terbesar pada korpus novel terdapat pada percobaan ADJP-NNx dan JJx-NNx (percobaan 2).

$$\begin{aligned}
 \text{Persentase peningkatan yang terjadi} &= ((5.4081 - 5.3876) / 5.3876) * 100\% \\
 &= (0.0205 / 5.3876) * 100\% \\
 &= +0.3796 \%
 \end{aligned}$$

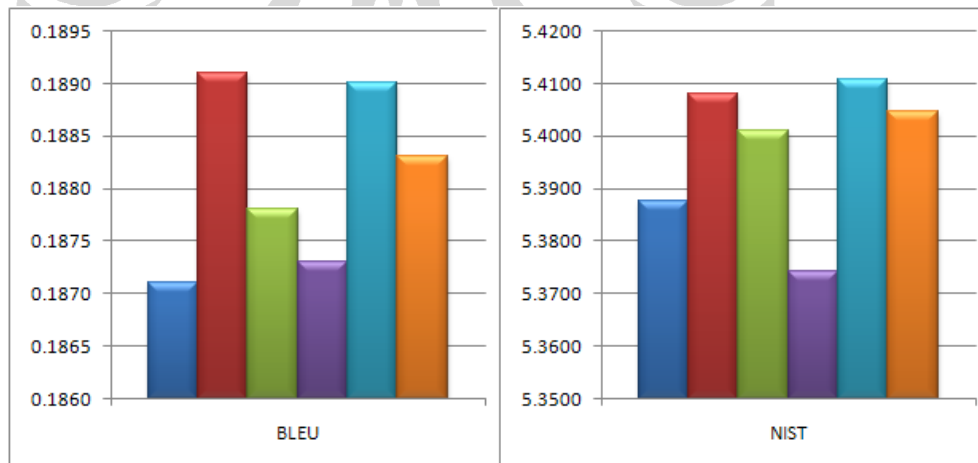
Perbandingan Tabel 5.22, ADJP-NNx, JJx-NNx, PRP\$-NNx direpresentasikan dalam bentuk grafik pada Gambar 5.8.



Gambar 5.7 Grafik Perbandingan Hasil Eksperimen *Mixed* dengan Hasil Eksperimen Lainnya Pada Korpus Bible

Keterangan:

- Baseline
- ADJP-NNx dan JJx-NNx
- ADJP-NNx dan PRPŞ-NNx
- ADJP-NNx
- JJx-NNx
- PRPŞ-NNx



Gambar 5.8 Grafik Perbandingan Hasil Eksperimen *Mixed* dengan Hasil Eksperimen Lainnya Pada Korpus Novel

Keterangan:

- Baseline
- ADJP-NNx dan JJx-NNx
- ADJP-NNx dan PRPŞ-NNx
- ADJP-NNx
- JJx-NNx
- PRPŞ-NNx

Dari Gambar 5.7, rancangan aturan *phrase and word reordering* yang dapat meningkatkan akurasi dan kualitas penerjemahan mesin penerjemah statistik pada korpus bible berdasarkan BLEU, adalah:

1. Penukaran frase sifat dengan kata benda dan kata sifat dengan kata benda (ADJP-NNx dan JJx-NNx)
2. Penukaran frase sifat dengan kata benda dan kata kepunyaan dengan kata benda (ADJP-NNx dan PRP\$-NNx)

Dari Gambar 5.7, tidak ada rancangan aturan *phrase and word reordering* yang dapat meningkatkan akurasi dan kualitas penerjemahan mesin penerjemah statistik pada korpus bible berdasarkan NIST.

Dari Gambar 5.8, rancangan aturan *phrase and word reordering* yang dapat meningkatkan akurasi dan kualitas penerjemahan mesin penerjemah statistik pada korpus novel berdasarkan BLEU dan NIST, adalah:

1. Penukaran frase sifat dengan kata benda dan kata sifat dengan kata benda (ADJP-NNx dan JJx-NNx)
2. Penukaran frase sifat dengan kata benda dan kata kepunyaan dengan kata benda (ADJP-NNx dan PRP\$-NNx)

Dari Gambar 5.7 dan Gambar 5.8 dapat dikatakan bahwa hasil eksperimen penggabungan dari *phrase reordering* dan *word reordering* cenderung tidak mengalami perubahan dan peningkatan nilai yang berarti daripada hasil eksperimen *word reordering*. Hasil peningkatan yang diperoleh *word reordering* masih lebih besar daripada penggabungan *phrase reordering* dan *word reordering*. Apabila dibandingkan dengan hasil eksperimen *phrase reordering*, hasil eksperimen penggabungan *phrase reordering* dan *word reordering* cenderung lebih baik dan lebih memberikan perubahan dan peningkatan nilai.

5.4.1 Analisis Kesalahan Eksperimen Arsitektur Sistem *Mixed*

Faktor yang menyebabkan terjadinya *error* dan penurunan kualitas dan akurasi pada arsitektur sistem *mixed* sama dengan faktor yang menyebabkan *error* dan

penurunan kualitas dan akurasi pada arsitektur sistem POS tagging (subbab 5.2.1) dan arsitektur sistem parsing (subbab 5.3.1).

5.5 Rangkuman Analisis Eksperimen

➤ Untuk proses *word reordering* pada arsitektur sistem POS *tagging* dan *mixed*:

- Pada kalimat dengan kemunculan kata sifat atau kata benda atau kata kepemilikan yang diikuti langsung dengan kata benda, proses *word reordering* cenderung memberikan peningkatan akurasi dan kualitas penerjemahan. Struktur kalimat Bahasa Inggris setelah ditukar menjadi lebih mirip dengan struktur kalimat Bahasa Indonesia. Hal tersebut membantu proses penerjemahan dari mesin penerjemah statistik.

Berikut adalah contoh kenaikan *score* pada segment 6500 novel. Sebelum dilakukan JJx-NNx *reordering* nilai BLEU adalah 0.4483. Setelah dilakukan JJx-NNx *reordering* nilai BLEU meningkat menjadi 0.4962.

S	your first question , teabing declared , his tone herculean .
SR	your question first , teabing declared , his tone herculean .
T	anda yang pertama kalinya , kata teabing , nada suaranya herculean .
TR	pertanyaan anda dulu , kata teabing , nada suaranya herculean .
R	pertanyaan pertama , kata teabing , nada suaranya seperti herkules .

- Proses penukaran nama orang yang terdiri dari 2 kata yang berurutan dapat menyebabkan penurunan akurasi dan kualitas dari mesin penerjemah statistik (subbab 5.2.1).
- Adanya kata – kata dengan pola yang seharusnya ditukar pada level frase mengalami penukaran pada proses *word reordering* sehingga menyebabkan penurunan akurasi dan kualitas dari mesin penerjemah statistik (subbab 5.2.1).
- Kesalahan *tagging* dari Stanford POS Tagger. Hal ini dapat menyebabkan *error* dalam proses *word reordering* dan penurunan akurasi dan kualitas dari mesin penerjemah statistik (subbab 5.2.1).

- Beberapa kata Bahasa Inggris pada rujukan yang seharusnya diterjemahkan menjadi Bahasa Indonesia, tidak diterjemahkan menjadi Bahasa Indonesia. Kata tersebut tetap ditulis dalam Bahasa Inggris. Hal ini menyebabkan penurunan akurasi dan kualitas dari mesin penerjemah statistik (subbab 5.2.1).

➤ Untuk proses *phrase reordering* pada arsitektur sistem *parsing* dan *mixed*:

- Pada kalimat dengan kemunculan frase sifat atau frase benda yang diikuti langsung dengan frase benda atau kata benda, proses *phrase reordering* cenderung memberikan peningkatan akurasi dan kualitas penerjemahan. Struktur kalimat Bahasa Inggris setelah ditukar menjadi lebih mirip dengan struktur kalimat Bahasa Indonesia. Hal tersebut membantu proses penerjemahan dari mesin penerjemah statistik. Berikut adalah contoh kenaikan *score* pada segment 338 novel. Sebelum dilakukan ADJP-NNx *reordering* nilai BLEU adalah 0.4297. Setelah dilakukan ADJP-NNx *reordering* nilai BLEU meningkat menjadi 0.4844.

S	after all , he had somehow obtained <u>very secret information</u> .
SR	after all , he had somehow obtained <u>information very secret</u> .
T	lagi pula , dia telah sangat obtained sangat informasi rahasia .
TR	lagi pula , dia sudah entah bagaimana obtained informasi yang sangat rahasia .
R	lagi pula , dia telah mendapatkan informasi yang sangat rahasia .

- Proses *phrase reordering* memiliki kelemahan yang menyebabkan penurunan akurasi dan kualitas penerjemahan. Penurunan akurasi dan kualitas penerjemahan tersebut terjadi pada penukaran pada pasangan frase yang masih memiliki *parent* yang sama dengan jarak yang jauh. (subbab 5.3.1)
- Kesalahan *parsing* struktur *tree* dari Stanford Parser. Hal ini dapat menyebabkan *error* dalam proses *phrase reordering* dan penurunan akurasi dan kualitas dari mesin penerjemah statistik. (subbab 5.3.1)

➤ Perbandingan hasil yang diperoleh pada penelitian ini dengan penelitian lainnya adalah:

- Penelitian Michael Collins pada Bahasa Jerman - Bahasa Inggris. Hasil eksperimen Collins menunjukkan adanya peningkatan nilai BLEU dari 0.252 menjadi 0.268, mengalami peningkatan 6.3492% (Collins, Koehn, & Kucerova, 2005).
- Penelitian Maja Popovic pada Bahasa Inggris – Bahasa Jerman dan pada Bahasa Spanyol – Inggris. Hasil eksperimen ditampilkan pada Tabel 5.23 dan Tabel 5.24. Maja Popovic menggunakan korpus EUROPARL dan standar penilaian BLEU.

Tabel 5.23 Hasil Eksperimen Nilai BLEU Bahasa Inggris - Jerman

BLEU		EUROPARL Data Set 1	Perubahan (%)	EUROPARL Data Set 2	Perubahan (%)
7K sentence	<i>Baseline</i>	0.150		0.154	
	Reorder Verb	0.155	+3.3333%	0.160	+3.8961%
700K sentence	<i>Baseline</i>	0.196		0.197	
	Reorder Verb	0.204	+4.0816%	0.208	+5.5837%

Tabel 5.24 Hasil Eksperimen Nilai BLEU Bahasa Spanyol - Inggris

BLEU		EUROPARL Data Set 1	Perubahan (%)	EUROPARL Data Set 2	Perubahan (%)
7K sentence	<i>Baseline</i>	0.196		0.200	
	Reorder Verb	0.218	+11.2244%	0.221	+10.5%
700K sentence	<i>Baseline</i>	0.274		0.274	
	Reorder Verb	0.284	+3.6496%	0.281	+2.5547%

Dari Tabel 5.23 dan Tabel 5.24, hasil eksperimen Maja Popovic menunjukkan adanya peningkatan nilai BLEU (Popovic, & Ney, 2006). Rata – rata peningkatan nilai BLEU yang diperoleh pada penerjemahan Bahasa Inggris - Jerman yang diperoleh dari korpus EUROPARL adalah 4.2236%. Rata – rata peningkatan nilai BLEU yang diperoleh pada penerjemahan Bahasa Spanyol - Inggris yang diperoleh dari korpus EUROPARL adalah 6.9821%.

- Penelitian Sangodkar pada Bahasa Inggris - India. Sangodkar menggunakan korpus EILMT dan IIT serta standar penilaian BLEU dan NIST. Hasil eksperimen ditampilkan pada Tabel 5.25.

Tabel 5.25 Hasil Eksperimen Nilai BLEU dan NIST Bahasa Inggris - India

BLEU		Data Set 1	Perubahan (%)	Data Set 2	Perubahan (%)	NIST	Data Set 1	Perubahan (%)	Data Set 2	Perubahan (%)
EILMT	<i>Baseline</i>	0.1488		0.1450		<i>Baseline</i>	4.7600		4.7287	
	Reorder Verb	0.1751	+17.6747%	0.1601	+9.43%	Reorder Verb	4.8539	+1.9726%	4.6923	-0.7697%
IIT	<i>Baseline</i>	0.0815		0.0842		<i>Baseline</i>	3.9036		4.2426	
	Reorder Verb	0.0836	+2.5766%	0.0853	+1.30%	Reorder Verb	3.7335	-4.3575%	4.0140	-5.3882%

Dari Tabel 5.25 percobaan hasil eksperimen Sangodkar menunjukkan adanya peningkatan pada nilai BLEU. Akan tetapi pada percobaan tersebut terdapat penurunan nilai NIST pada korpus EILMT pada data set 2 serta korpus IIT pada data set 1 dan data set 2 (Sangodkar, Vasudevan, & Damani, 2008).

Rata – rata peningkatan nilai BLEU yang diperoleh pada penerjemahan Bahasa Inggris - India yang diperoleh dari korpus EILMT adalah 13.5524%. dan pada korpus IIT adalah 1.9383%. Rata – rata peningkatan nilai NIST yang diperoleh pada penerjemahan Bahasa Inggris - India yang diperoleh dari korpus EILMT adalah 0.3007%. sedangkan untuk korpus IIT terjadi penurunan 4.87285%.

- Rangkuman hasil eksperimen yang diperoleh dan memberikan peningkatan nilai BLEU atau NIST pada penelitian ini dilaporkan pada Tabel 5.26.

Tabel 5.26 Rangkuman Hasil Eksperimen Nilai BLEU dan NIST pada Penelitian

BLEU Inggris - Indonesia		Nilai	Perubahan (%)	NIST Inggris - Indonesia		Nilai	Perubahan (%)
Bible	<i>Baseline</i>	0.1831		<i>Baseline</i>	5.3542		
	JJx-NNx	0.1843	+0.6554%	JJx-NNx	5.3622	+0.1494%	
	JJx-NNx dan NNx-NNx	0.1846	+0.8192%	JJx-NNx dan NNx-NNx	5.3580	+0.0710%	
	JJx-NNx dan PRP\$-NNx	0.1848	+0.9285%	JJx-NNx dan PRP\$-NNx	5.3573	+0.0579%	
	JJx-NNx, NNx-NNx, dan PRP\$-NNx	0.1849	+0.9831%	JJx-NNx, NNx-NNx, dan PRP\$-NNx	5.3493	-0.0915%	
	ADJP - NNx	0.1832	+0.0546%	ADJP - NNx	5.3598	+0.1046%	
	ADJP - NP	0.1835	+0.2185%	ADJP - NP	5.3564	+0.0411%	
	ADJP-NNx dan JJx-NNx	0.1843	+0.6554%	ADJP-NNx dan JJx-NNx	5.3476	-0.1233%	
ADJP-NNx dan PRP\$-NNx	0.1838	+0.3823%	ADJP-NNx dan PRP\$-NNx	5.3368	-0.3250%		
Novel	<i>Baseline</i>	0.1871		<i>Baseline</i>	5.3876		
	JJx-NNx	0.1890	+1.0155%	JJx-NNx	5.4108	+0.4306%	
	JJx-NNx dan NNx-NNx	0.1897	+1.3896%	JJx-NNx dan NNx-NNx	5.4211	+0.6218%	
	JJx-NNx dan PRP\$-NNx	0.1892	+1.1224%	JJx-NNx dan PRP\$-NNx	5.4166	+0.5383%	
	JJx-NNx, NNx-NNx, dan PRP\$-NNx	0.1906	+1.8707%	JJx-NNx, NNx-NNx, dan PRP\$-NNx	5.4317	+0.8185%	
	ADJP - NNx	0.1873	+0.1069%	ADJP - NNx	5.3741	-0.2506%	
	ADJP - NP	0.1872	+0.0534%	ADJP - NP	5.3741	-0.2506%	
	ADJP-NNx dan JJx-NNx	0.1891	+1.0689%	ADJP-NNx dan JJx-NNx	5.4081	+0.3796%	
ADJP-NNx dan PRP\$-NNx	0.1878	+0.3741%	ADJP-NNx dan PRP\$-NNx	5.401	+0.2487%		

Kotak yang diarsir hijau menandakan nilai yang lebih tinggi dari nilai *baseline*. Kotak yang diarsir biru menandakan nilai yang lebih rendah dari nilai *baseline*.

Pada korpus bible, rata – rata nilai peningkatan nilai BLEU yang diperoleh adalah 0.5871% dan untuk nilai NIST terjadi penurunan sebesar 0.0144%.

Pada korpus novel, rata – rata nilai peningkatan nilai BLEU yang diperoleh adalah 0.8751% dan untuk nilai NIST terjadi peningkatan sebesar 0.3170%.

- Rata – rata jumlah peningkatan dan penurunan yang terjadi pada penelitian ini jika dibandingkan dengan percobaan lain yang pernah dilakukan adalah

cenderung kecil (masih di bawah 1%). Hal ini dikarenakan aturan penerjemahan Bahasa Inggris-Indonesia menggunakan aturan MD-DM yang melibatkan penukaran kata yang jaraknya dekat (bersifat *local reordering*). Penukaran kata tersebut mirip dengan *reordering* yang sudah dimiliki oleh mesin penerjemah statistik sehingga proses *reordering* Bahasa Inggris-Indonesia dengan menggunakan aturan MD-DM tidak terlalu berpengaruh. Berbeda dengan aturan *reordering* pada bahasa lain (cth: Jerman - Inggris, Inggris - India) yang melibatkan penukaran kata yang jauh, penukaran kata yang jauh tersebut tidak bisa ditangani oleh *reordering* yang ada pada mesin penerjemah statistik. Oleh karena itu, aturan *reordering* pada bahasa lain tersebut memberikan dampak perubahan nilai BLEU dan NIST yang lebih besar.

- Hasil yang diperoleh pada penelitian ini mirip dengan hasil yang didapatkan dari penelitian Sangodkar pada Bahasa Inggris – India, yaitu terjadi peningkatan untuk nilai BLEU dan terdapat variasi peningkatan dan penurunan nilai NIST pada eksperimen.