# PENERJEMAH DUA ARAH BAHASA INDONESIA KE BAHASA DAERAH (KARO) MENGGUNAKAN TEKNIK *STATISTICAL MACHINE TRANSLATION* (SMT) SEBAGAI FITUR PADA SITUS WEB UNTUK MENINGKATKAN *WEB TRAFFIC*

**Adres Ginting[1], Nazori AZ[2]**

**Magister Ilmu Komputer Program Pascasarjana Universitas Budi Luhur**
[1]aginting@gmail.com, [2]nazori@budiluhur.ac.id

**ABSTRAK**

*Penerjemahan dengan teknik Statistical Machine Translation (SMT) dapat dilakukan dengan mengolah data dari kumpulan kalimat sumber dan terjemahannya, yang disebut parallel corpus. Metode SMT mengeliminasi kebutuhan akan ahli linguistik karena terjemahan dilakukan oleh sistem berdasarkan statistik dari parallel corpus tersebut. Penggunaan teknik SMT ini telah dicoba dalam berbagai bahasa di dunia dengan hasil cukup baik pada sejumlah penelitian, sehingga aplikasinya pada bahasa Indonesia dan bahasa daerah (Karo) diharapkan dapat menghasilkan terjemahan yang baik pula. Sumber parallel corpus yang digunakan dalam penelitian ini adalah kumpulan kalimat dari sebagian kitab Injil berbahasa Indonesia dan kitab Injil berbahasa Karo berjumlah masing-masing sekitar 4000 baris dan 90.000 kata, dan sebagai pembanding digunakan kumpulan kalimat terjemahan dan kumpulan sinonim kata kedua bahasa tersebut yang berasal dari nara sumber berjumlah masing-masing sekitar 6000 baris dan 10.000 kata. Dengan corpus tersebut diperoleh skor hasil pengujian terjemahan dengan parameter fluency sebesar 1,9 dan 1,8 dari skala 5. Hasil pengujian ini, yang cukup baik jika skor lebih dari 3, menunjukkan bahwa perlu ditambahnya jumlah kalimat (dan kata) yang baik pada parallel corpus. Sistem penerjemah bahasa daerah ini dibangun berbasis web untuk pemasukan teks sumber dan menampilkan teks keluaran (terjemahan) dengan kemampuan dua arah, yang dapat dipublikasikan lewat internet untuk akses oleh publik.*

**Kata kunci**: *Statistical Machine Translation* (SMT), *Parallel corpus, Common Group Interface* (CGI), *Web traffic*

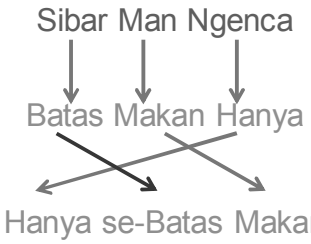
## 1. Pengantar

Pengaruh positif dari pernyataan pada Sumpah Pemuda tahun 1928 adalah keseragaman penggunaan bahasa Indonesia secara menyeluruh ke seluruh wilayah Indonesia. Sekolah-sekolah, pasar, bahasa pergaulan sehari-hari, kantor, transaksi bisnis, rumah ibadat, dan lain-lain, hampir semuanya menggunakan bahasa Indonesia sebagai bahasa utama.

Tetapi di lain pihak Indonesia memiliki keragaman bahasa dan budaya yang luar biasa, ada sebanyak 726 bahasa daerah dengan 719 bahasa daerah diantaranya yang masih aktif digunakan sehari-hari di seluruh pelosok Indonesia [1]. Diantara bahasa-bahasa daerah ini banyak yang terancam punah karena penggunanya yang semakin berkurang semakin hari [2], atau disebabkan juga oleh pengaruh asimilasi dengan bahasa utama yang lebih dominan. Bahasa-bahasa daerah di Indonesia termasuk dalam urutan tertinggi yang rawan punah dibandingkan bahasa dunia lainnya [3].

Bahasa daerah ini perlu dipelihara dan dilestarikan sebagai kekayaan budaya yang sangat bernilai di masa depan. Seperti juga kata pepatah bangsa yang besar adalah bangsa yang menghargai budayanya.

Penerjemahan dari suatu bahasa ke bahasa lain dengan cara terjemahan kata demi kata (*interlinear translation*) tidak selalu pas, terkadang menghasilkan terjemahan yang aneh, kaku, dan membuat hilangnya arti dan maksud yang terkandung dari suatu kalimat. Sebagai contoh terjemahan kata-per-kata dari kalimat "Sibar man ngenca" dalam bahasa Karo akan berubah menjadi "Sampai makan hanya" dalam bahasa Indonesia yang terasa aneh dan membingungkan, terjemahan yang benarnya adalah "Hanya se-batas makan". Lihat pada Gambar 1 berikut ini bahwa susuan kata-kata menjadi salah, walaupun terjemahan kata-per-kata nya sudah cukup baik.



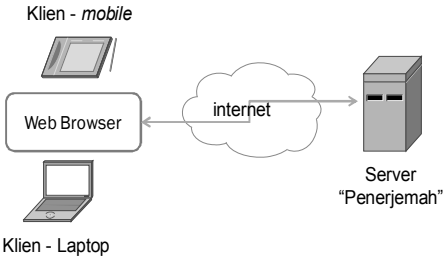

Gambar 1. Pengurutan kata yang salah dalam kalimat

Jadi terjemahan yang lebih baik tidak cukup hanya kata-per-kata, tetapi juga susunan kata dan konteksnya [4], seperti pada contoh di baris kedua Gambar 1 di atas yang tidak memperhatikan urutan kata-kata.

## 2. Tujuan Penelitian

Tujuan dari penelitian ini adalah memanfaatkan *Statistical Machine Translation* (SMT) untuk penerjemah bahasa secara dua arah, khususnya bahasa Indonesia ke bahasa daerah (Karo), yang dapat diakses melalui web *browser* yang pada akhirnya dapat dipublikasikan melalui sarana internet yang secara online dapat diakses secara luas.

Dalam penelitian ini dibangun sebuah sistem penerjemah (*server*) bahasa yang secara otomatis melakukan penerjemahan terhadap teks masukan yang diberikan kepadanya seperti digambarkan berikut ini.



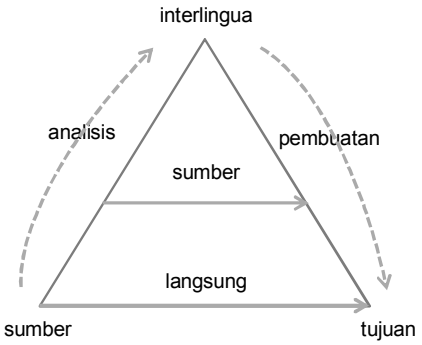

Gambar 2. Server Penerjemah Bahasa

Pada Gambar 2 di atas, klien berupa PC komputer, *laptop* atau perangkat *mobile* lainnya yang dapat menjalankan aplikasi *browser* merupakan klien untuk server penerjemah.

## 3. Sekilas *Machine Translation* (MT)

Ide awal dari *Machine Translation* (MT) diperkenalkan pertama kali oleh Warren Weaver [4] pada sekitar tahun 1949 dengan memorandumnya "*Translation*", dia adalah seorang ilmuwan dan ahli matematika dari Amerika Serikat. Idenya adalah menggunakan komputer digital untuk menerjemahkan dokumen dengan hasil berupa bahasa yang alami, termasuk didalamnya ide untuk penggunaan teori informasi (*noisy channel*) dari Claude Shannon.

Beberapa kriteria dapat digunakan untuk mengklasifikasikan pendekatan MT, namun yang paling sering digunakan adalah berdasarkan tingkatan analisis linguistik (dan pembangkitannya) yang dibutuhkan sistem untuk melakukan prosedur penerjemahan yang dapat digambarkan dengan paramida MT pada Gambar 3 berikut.




Gambar 3. Piramida *Machine Translation* [5]

Secara umum, pada bagian terbawah dari piramida mewakili sistem yang sama sekali tidak melakukan analisis kalimat sumber dalam rangka menghasilkan kalimat target (tujuan). Biasa disebut terjemahan langsung (*direct translation*) dari kata-per-kata.

Di piramida atasnya, sistem yang melakukan analisis sederhana berdasarkan aturan morfologis dan sintaksis, yang disebut sebagai *Transfer-based translation* atau terjemahan dengan cara transfer. Pada pendekatan ini dilakukan proses dengan analisis tata-bahasa dari sumber dan tujuan, kemudian aturan dibuat untuk konversi teks menjadi struktur tertentu dan pembangkitan teks tujuan dari teks sumber. Umumnya aturan tersebut dibuat secara manual sehingga membutuhkan bantuan tenaga manusia yang mengerti komparasi tata-bahasa antara sumber dan tujuan. Pendekatan ini banyak digunakan untuk MT pada tahun 1980-an, walaupun hanya pada domain yang terbatas.

Tingkat tertinggi pada piramida MT tersebut adalah sistem yang melakukan analisis semantik dari kalimat sumber dan melakukan penerjemahan berdasarkan aturan semantik yang terwakili dari sumber dan tujuan. Pendekatan yang dilakukan dengan analisis mendalam dari kalimat sumber untuk mendapatkan semantiknya yang disebut *Interlingua*. Bahasa konseptual ini, yang perlu dibangun, memiliki keuntungan bahwa jika sekali arti dari sumbernya diketahui maka secara teori dapat dituliskan dalam berbagai bahasa tujuan selama mesin pembangkitan tersedia untuk tiap bahasa tujuannya. Disamping sulitnya membuat konseptual bahasa, untuk membuat mesin yang mengerti kalimat sumber sebelum diterjemahakan juga merupakan hal yang berat, terutama untuk susunan bahasa informal yang seringkali tidak mengikuti aturan umum.

Pendekatan lain yang kontras dengan 3 pendekatan di atas adalah dengan menggunakan *corpus*. Sistem ini mengekstrak informasi yang dibutuhkan untuk membangkitkan terjemahan dari *parallel corpus* yang meliputi kumpulan kalimat-kalimat yang telah diterjemahkan oleh orang sebelumnya.

Diantara pendekatan berbasis corpus yang bermunculan pada tahun 1990-an adalah pendekatan berbasis contoh (*Example-bases* MT - EBMT) dan berbasis statistik (*statistical* MT – SMT), walaupun perbedaan antar keduanya terus jadi perdebatan. Pada EBMT penggunaan *parralel corpora* sebagai database dari contoh terjemahan, yang dibandingkan dengan kalimat masukan dalam rangka melakukan terjemahan. Dalam SMT, proses ini dilakukan dengan fokus pada parameter statistik dan kumpulan model translasi dan model bahasa. Walaupun pada mulanya masih menggunakan pendekatan kata-per-kata, yang dapat diklasifikasikan pada terjemahan langsung, tetapi pada saat ini sudah banyak mesin yang menggunakan analisis linguistik dalam tingkatan tertentu dalam SMT sehingga SMT sudah ada pada jajaran atas piramida MT tersebut [5].

### 3.1 *Statistical Machine Translation* (SMT)

*Statistical* MT (SMT) yang murni kemudian diperkenalkan pada tahun 1991 oleh para periset ahli pada Thomas J. Watson Research Center – IBM. Dengan kemampuan komputer yang telah meningkat pesat pada sekitar tahun 90-an maka dimungkinkan untuk melakukan pengembangan dengan teknik statistik pada sistem komputer. Hasil riset tim IBM ini memberikan kontribusi signifikan dalam peningkatan ketertarikan dan optimisme para peneliti dan komunitasnya pada kemampuan mesin penerjemah dengan komputer.

*Statistical machine translation (SMT)* adalah suatu paradigma dari mesin penerjemah dimana penerjemahan dilakukan berbasis model statistik dengan parameter-parameter yang diturunkan dari analisis *parallel corpus*. Pendekatan statistik berbeda dengan pendekatan berbasis-aturan (*rule based*) dan berbeda dari translasi berbasis contoh kalimat.

Penerjemahan dengan metode *Statistical Machine Translation* (SMT) menghasilkan terjemahan yang lebih baik dibandingkan dengan hanya terjemahan kata demi kata (*interlinear translation*), dengan syarat kualitas *parallel corpus* (padanan kalimat-kalimat bahasa sumber dan bahasa tujuan) yang dimasukkan ke dalam sistem

mempunyai kualitas baik dan cukup banyak jumlahnya [6].

Ide dibalik SMT berbasis pada asumsi bahwa setiap kalimat *e* pada bahasa tujuan merupakan sebuah kemungkinan translasi dari kalimat *f* yang diberikan dari bahasa sumber. Perbedaan utama antara dua translasi yang berbeda dari sebuah kalimat adalah karena perbedaan probabilitas yang diberikan ke masing-masing, yang mana probabilitas tersebut akan dipelajari dan diekstrak dari *parallel corpus* yang ada.

Dengan pendekatan bahwa teks yang diterjemahkan berdasarkan distribusi probabilitas *Pr(e | f)* bahwa teks *e* pada bahasa tujuan adalah translasi dari teks *f* pada bahasa sumber, yang dapat dituliskan sebagai:

$$\hat{e} = \arg\max_{e} Pr(e \mid f)$$

Masalah pemodelan dari distribusi probabilitas *Pr(e | f)* telah dilakukan pendekatan dengan berbagai cara. Salah satu pendekatan intuitif adalah dengan teorema Bayes, bahwa:

$$Pr\,(e \mid f) \propto Pr(f \mid e)\,.\,Pr(e)$$

dimana model translasi *translation model Pr(f | e)* adalah probabilitas teks sumber adalah translasi dari teks tujuan, dan model bahasa *language model Pr(e)* adalah probabilitas melihat teks bahasa tujuan. Hasil pemecahan ini menjadi sangat menarik karena memisahkan masalah menjadi dua sub-masalah. Yakni menemukan translasi terbaik dari $\hat{e}$ dilakukan dengan mengambil salah satu yang memberikan probabilitas tertinggi (*arg max*), menjadi:

$$\hat{e} = \arg\max_{e} Pr(f \mid e) \cdot Pr(e)$$

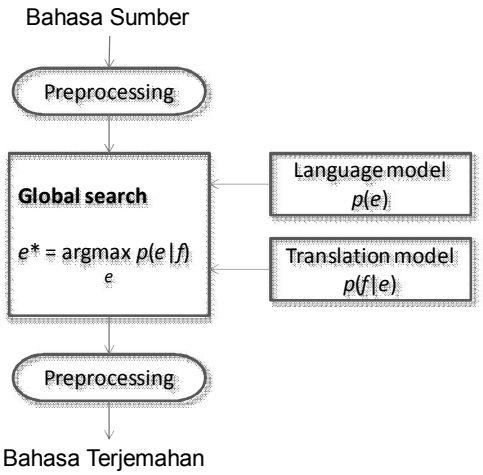
Dimana model bahasa, yang biasanya diimplementasikan menggunakan *N-grams*, telah diaplikasikan dengan sukses pada pemrosesan suara dan bidang-bidang lainnya, model translasi diperkenalkan dengan variabel *a* yang tersembunyi untuk memperkirakan relasi antara kata-kata pada tiap bahasa, dengan perumusan:

$$Pr(f \mid e) = \sum_{a} Pr(f, a \mid e) = Pr(J \mid e) \prod_{j=1}^{J} Pr(a_j \mid f_1^{j-1}, a_1^{j-1}, e) \cdot Pr(f_j \mid f_1^{j-1}, a_1^{j}, e)$$

Dimana $f_j$ adalah kata dalam posisi *j* dari teks sumber *f*, *J* adalah panjang dari teks (jumlah kata), dan $a_j$ adalah pelurusan/penyesuaian untuk kata $f_j$, misalnya posisi dari teks tujuan *e* dimana kata tersebut sesuai dengan peletakan $f_j$ . Parameter-parameter dari model atau probabilitasnya didapatkan dari *parallel corpus* [5].

Untuk dapat melakukan pelatihan *training* pada parameter-parameter yang sangat banyak maka dilakukan dengan algoritma EM (ekspektasi-maksimalisasi *Expectation – Maximalitation*) model kompleks. Model-model ini ini secara luas dikenal sebagai Lima Model IBM atau *Five IBM Model.* Dalam implementasinya *toolkit* yang banyak digunakan untuk *training* model IBM 1 – 5 adalah *Giza++* [6].

Proses dekoding (*decoding*) secara umum dapat digambarkan juga seperti pada Gambar 4 berikut ini.




Gambar 4. Diagram Pemrosesan dalam SMT

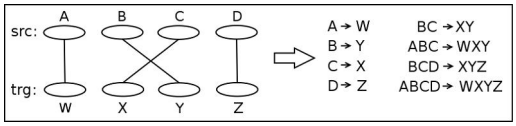
Dari gambaran di atas, untuk implementasi yang sempurna tentunya akan membutuhkan proses pencairan (*search*) yang luar biasa banyak dengan mencari semua teks *e** pada bahasa asli. Melakukan pencarian yang efisien adalah tugas dari dekoder (*decoder*) mesin translasi yang menggunakan metode heuristik dan metode-metode lainnya untuk membatasi ruang pencarian yang dapat

ditangani dan secara bersamaan sambil tetap menjaga kualitas yang dapat diterima.

Model Bahasa *Language Model* adalah sumber pengetahuan terpenting dalam SMT. Model bahasa standar *n-gram* memberikan probabilitas untuk hipotesis dalam bahasa tujuan yang terkondisikan dalam catatan konteks sebelumnya sejumlah n-1 kata-kata [7]. Model Bahasa biasanya diperkirakan dengan penghalusan model *n-gram*, dan pendekatan yang serupa juga dilakukan pada model translasi, tetapi ditambah dengan kompleksitas lain yang dikarenakan oleh panjang kalimat yang berbeda dan urutan-urutan kata pada bahasa.

Model Translasi (*Translation Model*) dapat menggunakan beberapa jenis pendekatan yaitu: model translasi berbasis kata, model translasi berbasis frase dan model translasi berbasis keduanya (kombinasi). Model translasi berbasis frase saat ini lebih dipilih karena pada model translasi tingkat kata menghilangkan banyak konteks lokal selama proses translasi.

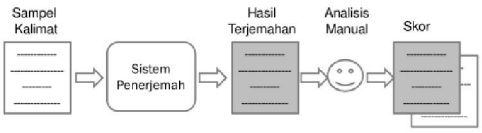
Dimana frase dua-bahasa didefinisikan sebagai pasangan sembarang dari frase sumber dan frase tujuan yang memiliki kata-kata berurutan dan konsisten dengan matriks penghubung kata. Berdasarkan kriteria ini, maka setiap rangkaian kata-kata sumber yang berurutan dan kata-kata tujuan yang berurutan yang saling terhubung satu sama lain dan bukan terhubung pada token lain dalam kalimat menjadi sebuah frase. Pernyataan di atas digambarkan pada Gambar 5 berikut ini, dimana ada delapan frase yang berbeda yang diekstrak dan bahwa AB → WY tidak termasuk yang diekstrak, sesuai dengan definisi tersebut.




Gambar 5. Ekstrak frase dari kata tertentu dari pasangan kalimat

### 3.2 *Evaluasi Machine Translation*

Mengevaluasi suatu *Machine Translation* secara otomatis adalah hal yang sangat sulit, biasanya yang dilakukan adalah mengukur dengan melihat kesamaan dari translasi hipotesis dan referensi translasi biasa, yang mewakili terjemahan yang diharapkan. Kenyataannya bahwa ada beberapa alternatif terjemahan untuk kalimat masukan turut menambah kompleksitas pekerjaan ini. Semakin dekat hasil terjemahan ke bahasa alami (bahasa manusia) maka bisa dianggap kualitasnya lebih baik, tetapi secara teoritis kita tidak dapat menjamin bahwa ketiadaan korelasi dengan referensinya berarti kualitas buruk selama kita punya semua kemungkinan terjemahan yang benarnya. Proses evaluasi sistem penerjemah digambarkan berikut ini.




Gambar 6. Proses Pengujian Sistem

Matrik evaluasi secara manual memerlukan keterlibatan si pengevaluasi dalam tingkatan tertentu untuk mendapatkan skor penilaian hasil terjemahan yang baik. Secara internasional umumnya cenderung diukur dengan faktor kecukupannya (*adequacy*) dan kefasihannya (*fluency*) [5]. *Fluency* mengindikasikan seberapa alami hasil terjemahan kepada si pengevaluasi yang fasih berbahasa hasil terjemahannya, dengan skor ditentukan sebagai berikut:

5 sempurna (*for Flawless*),
4 bagus (*for Good*),
3 tidak natif (*for Non-native*),
2 tidak pas (*for Disfluent*), dan
1 tidak memadai (*for Incomprehensible*).

Sedangkan *adequacy* diukur setelah itu, dan si pengevaluasi harus memutuskan seberapa banyak informasi yang berhasil dipindahkan ke hasil terjemahannya, dengan skor nilai:

5 seluruh informasi (*for all of the information*),
4 sebagian besar informasi (*for most of the information*),
3 banyak informasi (*for much of the information*),
2 sedikit iinformasi (*for little information*), dan
1 tidak ada satupun di atas (*for none of it*)

## 4. Tinjauan Studi

Penerjemahan dengan metode statistik lebih baik hasilnya dibandingkan dengan pendekatan berbasis aturan (*rule-based*) dalam berbagai evaluasi yang telah dilakukan oleh para peneliti. Demikian juga sistem ini lebih kebal terhadap kesalahan tata-bahasa karena dia tidak melakukan analisis yang mendalam pada kalimat sumber, melainkan sistem mencari hipotesis translasi yang mungkin dari kalimat sumber yang diberikan pada bahasa tujuan, dengan mengasumsikan kalimat tersebut adalah benar.

Kelebihan dari SMT adalah bahwa teknik pengembangannya, yang telah dilakukan pada sejumlah bahasa utama di Eropa seperti pasangan bahasa-bahasa Inggris – Perancis – Jerman – Spanyol, yang secara teori relatif sama untuk diaplikasikan ke pasangan bahasa-bahasa lainnya selama terdapat *parallel corpus* untuk bahan training sistem yang cukup memadai jumlah dan kualitasnya [5]. Untuk bahasa-bahasa di Asia seperti pasangan bahasa Inggris dan bahasa China, Korea, Jepang dan Arab telah diuji-cobakan dan menunjukkan hasil yang meningkat seperti diuraikan pada [5], [8]. Berikut ini adalah daftar sejumlah penelitian SMT pada Tabel 1 yang pernah dilakukan sebelumnya untuk beberapa bahasa-bahasa di Eropa, dengan jumlah kalimat dan kata *parallel corpus* yang digunakannya.

Tabel 1. Daftar Penelitian Bahasa-bahasa Eropa dengan SMT [9]

| Parallel Corpus (L1-L2) | Sentences | L1 Words | English Words |
|---|---|---|---|
| Bulgarian-English | 226,768 | - | 6,011,944 |
| Czech-English | 462,351 | 10,573,983 | 12,296,772 |
| Danish-English | 1,785,775 | 46,102,455 | 48,833,481 |
| German-English | 1,739,154 | 45,607,269 | 47,978,832 |
| Greek-English | 1,064,544 | - | 30,325,647 |
| Spanish-English | 1,786,594 | 51,551,485 | 49,411,045 |
| Estonian-English | 469,622 | 9,318,986 | 12,452,336 |
| Finnish-English | 1,742,553 | 34,123,013 | 47,601,416 |
| French-English | 1,825,077 | 54,568,499 | 50,551,047 |

Pada penelitian lainnya dengan menggunakan bahasa China ke bahasa Inggris dengan jumlah baris kalimat *parallel corpus* sebanyak 20 ribu baris dengan berbagai model diperoleh hasil evaluasi manual Tabel 2 seperti pada tabel berikut.

Tabel 2. Contoh hasil evaluasi dari bahasa China ke Inggris [5]

| Fluency | | Adequacy | |
|---|---|---|---|
| ITC-irst | 3.15 | MIT/AF | 2.71 |
| RWTH | 3.04 | ITC-irst | 2.65 |
| CMU | 2.88 | RWTH | 2.63 |
| ATR-C3 | 2.86 | UPCph | 2.52 |
| **UPC** | 2.82 | IBM | 2.51 |
| EDINBG | 2.81 | **UPC** | 2.44 |
| MIT/AF | 2.79 | EDINBG | 2.33 |
| UPCph | 2.78 | ATR-C3 | 2.31 |
| IBM | 2.77 | NTT | 2.09 |
| USC-ISI | 2.32 | CMU | 1.95 |
| NTT | 1.97 | USC-ISI | 1.90 |

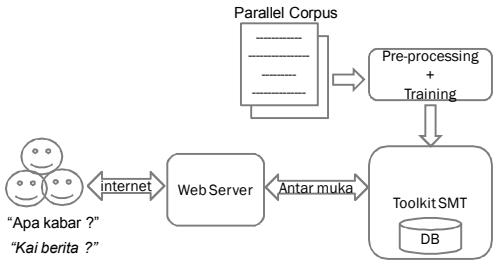
Dari tabel tersebut terlihat bahwa dengan model IBM diperoleh *fluency* 2,82 yang berada di antara skor 2 (tidak pas - *disfluent*) dan 3 (tidak natif - *non-native*). Sedangkan nilai *adequacy* 2,44 ada di antara skor 2 (sedikit informasi - *little information*) dan 3 (banyak informasi - *for much of the information*). Skor yang tidak terlalu tinggi ini disebabkan oleh faktor urutan kata atau *word order* yang berbeda antara bahasa sumber dan tujuan [5].

Pada penelitian ini akan dilakukan teknik SMT terhadap obyek bahasa yang berbeda dari penelitian-penelitian tersebut di atas, yaitu bahasa Indonesia dan bahasa Karo.

## 5. Implementasi

Membangun suatu sistem SMT merupakan pekerjaan yang sangat kompleks, namun dengan adanya *toolkit open source* yang tersedia akan sangat membantu riset dalam bidang ini. *Toolkit* open source *Moses* dikembangkan dari sistem *Pharaoh* di *University of Edinburgh*, dengan ditambahkan berbagai komponen utama lainnya. Moses merupakan sistem SMT yang lengkap, termasuk didalamnya komponen *training*, *tuning* dan *decoding* [10]. Moses menyediakan dua jenis model translasi yaitu: berbasis frase *phrase-based* dan berbasis pohon *tree-based*, kemudian ditambahkan

juga *factored translation models* yang memungkinkan integrasi dari informasi linguistik dan informasi lainnya pada tingkatan kata [9]. *Toolkit* ini dibangun dengan bahasa pemrograman komputer *C++* dan *Perl*.



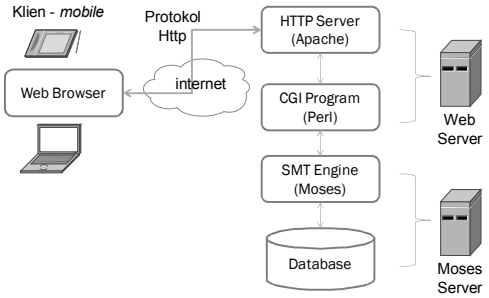

Gambar 7. Diagram Sistem Penerjemah

Data untuk penelitian ini berasal dari data primer. Data primer yang dijadikan sebagai *parallel corpus* didapat sebagian kitab Injil yaitu bagian Matius, Markus, Roma, Ibrani, Galatia dan Wahyu yang total berjumlah 4.130 ayat - baris kalimat pada Injil bahasa Indonesia dan 4.130 ayat - baris kalimat pada Injil bahasa Karo. Sumber kedua *parallel corpus* sebagai pembanding adalah kumpulan terjemahan kalimat kedua bahasa dan sinonim kata dari nara sumber yang secara keseluruhan berjumlah 5.794 baris kalimat/kata dalam bahasa Indonesia dan 5.794 baris kalimat/kata dalam bahasa Karo. Untuk bahan pengujian hasil terjemahan bahasa digunakan *systematic sampling* sejumlah lebih dari 2,5% atau 118 baris dari *parallel corpus* kitab Injil ditambah kalimat bebas digunakan sebagai sampel.

### 5.1 Perancangan Sistem Penerjemah Bahasa

*Engine* SMT yang digunakan dibangun dari *toolkit Moses* seperti diuraikan di atas perlu dibuatkan suatu antar-muka ke sisi user yang meliputi pembuatan tampilan halaman depan pada situs web. Sistem berbasis web yang dibuat menggunakan perangkat bantu (*toolkit*) web server (*Apache* versi 2), untuk bahasa pemrograman komputer berbasis web yang digunakan adalah *Perl* versi 5.12.4. Hal ini menyesuaikan dengan bahasa pemrograman yang digunakan oleh *toolkit* SMT tersebut.

Rancangan implementasi yang dibuat dapat dilihat pada Gambar 8 berikut ini.

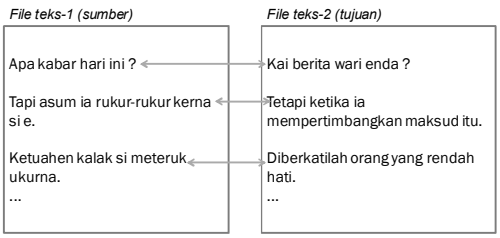



Gambar 8. Diagram hubungan Klien – *Web Server – Engine* SMT

Keseluruhan sistem ini di-set dan dijalankan pada satu server dengan konfigurasi yang minim untuk tujuan prototipe saja pada penelitian ini.

### 5.2 Persiapan *Parallel Corpus*

*Parallel corpus* dibuat dalam bentuk file teks dengan susunan seperti digambarkan berikut ini.




Gambar 9. Susunan File Teks – *Parallel Corpus*

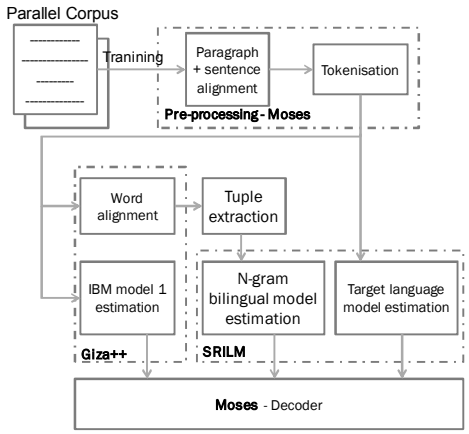
File teks yang digunakan untuk *toolkit Giza++, SRILM dan Moses* disyarakat dalam format *utf-8*.

Dari sebagian kitab Injil yang akan dipakai yaitu bagian Matius, Markus, Roma, Ibrani, Galatia dan Wahyu yang total berjumlah 4.130 ayat - baris kalimat yang terdiri atas sejumlah 86.929 kata pada Injil bahasa Indonesia dan 4.130 ayat - baris kalimat dan 97.270 kata pada Injil bahasa Karo. Sedangkan sumber kedua *parallel corpus* sebagai pembanding adalah kumpulan terjemahan kalimat kedua bahasa dan sinonim kata dari nara sumber yang secara

keseluruhan berjumlah 5.794 baris kalimat dengan 10.042 kata dalam bahasa Indonesia dan 5.794 baris kalimat dengan 9.336 kata dalam bahasa Karo.

### 5.3 Pemrosesan *Parallel Corpus*

*Parallel corpus* dalam file teks yang telah dipersiapkan akan diproses dengan dimulai dari *pre-processing* (pra-proses) berupa: *Tokenize training data*, *Filter out long sentences* dan *Lowercase training data.* Proses selanjutnya adalah membangun model bahasa (*language modelling*) dan *training* pada sistem. Pemrosesan (*training*) *parallel corpus* sampai dengan dapat digunakan oleh *decoder Moses* dapat dilihat pada berikut ini.

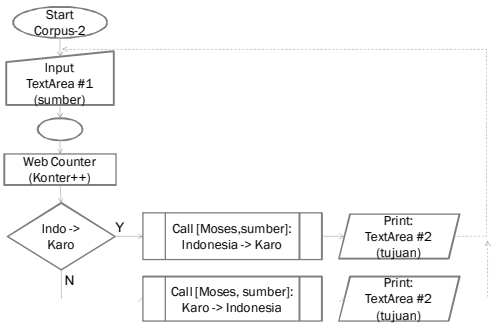



Gambar 10. Pemrosesan *Parallel Corpus*

Proses seperti di atas perlu dilakukan untuk setiap parallel corpus yang digunakan, dan untuk mendapatkan proses penerjemahan dua arah maka dibuat directory terpisah untuk masing-masing arah terjemahan.

### 5.4 Pembuatan Tampilan dan Program Antar Muka

Diagram alir berikut ini memfasilitasi user memasukkan kalimat sumber pada *TextArea #1,* kemudian memilih pilihan "Indo → Karo" untuk terjemahan kalimat bahasa Indonesia ke bahasa Karo atau pilihan "Karo → Indo" untuk terjemahan kalimat bahasa Karo ke bahasa Indonesia. Setelah itu user mengklik tombol ">> Terjemahkan !", maka program melalukan proses *Call* ke *Moses* dengan mem-*passing* parameter *TextArea #1* sebagai kalimat sumber.




Gambar 11. Diagram Alir – *Parallel Corpus* - 2

Perintah untuk *Call* ke *Moses* seperti pada dikutip berikut ini.

```
@ARGV = param('sumber');
my $terjemahan = `echo @ARGV | /home/adres/mosesdecoder/dist/bin/moses -f /home/adres/transkaro/data/work/model/moses.ini 2> /dev/null` or die "Couldn't execute command: $!";
```

Hasil dari proses *Call* tersebut berupa teks kalimat terjemahan *($terjemahan)* yang akan ditampilkan dengan perintah *print* pada area *TextArea #2*, seperti dapat di lihat di bawah ini.

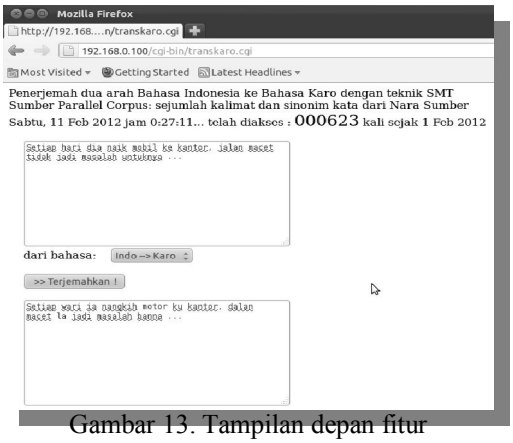
```
print p, "     ", textarea('trans', $terjemahan , 10, 50);
```

Fungsi *Web Counter* pada program antar-muka tersebut adalah untuk menghitung jumlah akses oleh user yang tiap kali menjalankan fungsi penerjemah, dengan variabel *konter++* akan bertambah satu setiap kali user mengklik tombol ">> Terjemahkan !".

Rancangan tampilan halaman muka dari situs web dengan akses simulasi ke http://transkaro.net/ kemudian di-link dengan halaman penerjemah.


Gambar 12. Tampilan halaman depan penerjemah

Tampilan untuk fitur penerjemah dua arah dengan akses simulasi ke link http://transkaro.net/cgi-bin/transkaro.cgi dapat dilihat seperti gambar 13 di bawah ini untuk penerjemahan dari bahasa Indonesia ke bahasa Karo. Fungsi WebCounter menampilkan jumlah penerjemahan yang telah dilakukan sejak situs diluncurkan, dalam contoh di bawah pada Gambar 13 menunjukkan angka 623 kali dan 618 kali. Dengan WebCounter ini dapat digunakan sebagai salah satu instrumen dalam menghitung *web traffic* yang timbul, dimana penambahan angkanya menandakan permintaan user untuk melakukan penerjemahan.




Gambar 13. Tampilan depan fitur penerjemah Indo → Karo

Sedangkan pada Gambar 14 berikut ini adalah untuk fungsi penerjemahan dari bahasa Karo ke bahasa Indonesia.


Gambar 14. Tampilan depan fitur penerjemah Karo → Indo

## 6. Interpretasi Hasil Pengujian

### 6.1 Pengujian Hasil *Parallel Corpus* Kitab Injil

Pada saat dilakukan *pre-processing* dari corpus kitab Injil terdapat banyak baris yang tidak memenuhi syarat sehingga dibuang, seperti ditunjukkan hasil di bawah ini:

```
root@ubuntu:/home/adres/trans-injil-karo/data#
/home/adres/mosesdecoder/scripts/training/clean-
corpus-n.perl
work/corpus/mix01.tok fr en
work/corpus/mix01.clean 1 40

clean-corpus.perl: processing
work/corpus/mix01.tok.fr & .en to
work/corpus/mix01.clean, cutoff 1-40

        Input sentences:  4130
        Output sentences:  3581
```

Persiapan *pre-processing* di atas yang menunjukkan ada sejumlah 4130 (*Input sentences)* – 3581 (*Output sentences*) = 549 baris corpus yang terbuang, atau hampir 13% tidak dapat masuk sebagai *corpus*. Jumlah yang terlalu besar, untuk itu perlu dilakukan pembenahan awal pada ayat/kalimat kitab Injil sebelum digunakan sebagai *corpus*. Pembenahan yang dapat dilakukan adalah pemisahan kalimat-kalimat dalam satu ayat menjadi sejumlah baris, dengan tetap memperhatikan ke-paralel-an dari file sumber dan file tujuannya.

Hasil percobaan menunjukkan bahwa banyak terjemahan yang masih menggunakan kata bahasa sumber (Karo) karena *corpus* yang digunakan dari kitab Injil tidak selalu paralel atau sesuai dengan posisi kalimat pada terjemahannya, mengakibatkan sistem salah dalam menemukan padanan terjemahan yang tepat. Demikian juga dan contoh terjemahannya masih sangat kurang jumlahnya sehingga sistem kekurangan referensi statisktik. Masalah lainnya adalah karena dalam satu ayat seringkali terjemahan bahasanya dilakukan dengan penjelasan tambahan dengan konteks budaya Karo yang berbeda dengan budaya Indonesia secara umum, sehinga ke-paralel-an kalimat terjemahan pada *corpus* juga sering kali tidak terjadi. Hal ini dapat dilihat pada contoh berikut kata "mereka" dalam bahasa Indonesia diasosiasikan dengan begitu

banyak kata dan frase lain yang tidak sesuai dalam bahasa Karo.

Statistik pada phrase-table:
File = /home/adres/trans-injil-karo/data/work/model/phrase-table.gz
...
*mereka ||| kalak jerusalem enda ||| 1 0.209069 0.000569476*
*mereka ||| kalak jerusalem ||| 1 0.209069 0.000569476*
*mereka ||| kalak kemamangen ||| 1 0.209069 0.000569476*
*mereka ||| kalak maka kam ||| 0.2 0.209069 0.000569476*
*mereka ||| kalak maka ||| 0.2 0.209069 0.000569476*
*mereka ||| kalak meteh maka kam enggo malem ||| 0.25 0.209069*
*mereka ||| kalak meteh maka kam enggo ||| 0.25 0.209069 0.000569476*
*mereka ||| kalak meteh maka kam ||| 0.25 0.209069 0.000569476*
*mereka ||| kalak meteh maka ||| 0.25 0.209069 0.000569476*
*mereka ||| kalak meteh ||| 0.25 0.209069 0.000569476*
*mereka ||| kalak ndai , ||| 0.05 0.161104 0.000569476*
*mereka ||| kalak ndai kerina ||| 1 0.209069 0.000569476*
*mereka ||| kalak ndai penungkunen enda , ||| 1 0.209069 0.000569476*
*mereka ||| lang-lang ia ||| 0.333333 0.194865 0.000569476*
*mereka ||| lawes ia ||| 0.04 0.194865 0.00113895*
*mereka ||| lawit e ||| 0.142857 0.0890553 0.000569476 4*
*mereka ||| lewi ||| 0.1 0.0833333 0.000569476 0.0005552*
...

Bandingkan dengan kata "mereka" pada tabel frase yang menggunakan *corpus* dari nara sumber yang lebih terpelihara ke-paralel-an corpus-nya seperti di bawah ini. Dapat dilihat bahwa kata "mereka" diasosiasikan dengan benar.

Statistik pada phrase-table:
File = /home/adres/transkaro/data/work/model/phrase-table.gz
...
*. mereka segera ||| . kalak e minter ||| 1 0.0418304 1 0.210605 2.718*
*. mereka pun ||| . kalakenda pe ||| 1 0.0194341 1 0.506536 2.718*
*. mereka ||| . kalakenda ||| 0.25 0.0303657 1 0.981413 2.718*
*. mereka ||| . kalak e ||| 0.25 0.0585625 0.5 0.252726 2.718*
...

## 6.2 Pengujian Hasil *Parallel Corpus* Nara Sumber

Pada saat dilakukan *pre-processing* terhadap *corpus* kumpulan kalimat dan sinonim kata dari nara sumber terlihat bahwa hampir semua baris pada *corpus* dapat diterima, seperti ditampilkan di bawah ini.

*root@ubuntu:/home/adres/transkaro/data#*
*/home/adres/mosesdecoder/scripts/training/clean-corpus-n.perl*
*work/corpus/mix01.tok fr en*
*work/corpus/mix01.clean 1 40*

*clean-corpus.perl: processing*
*work/corpus/mix01.tok.fr & .en to*
*work/corpus/mix01.clean, cutoff 1-40*

*Input sentences: 5794 Output sentences: 5785*

Dari hasil *pre-processing* bahwa pada saat training seperti di kutip di atas menunjukkan bahwa ada sejumlah 5794 (*Input sentences)* – 5785 (*Output sentences)* = 9 baris corpus yang terbuang, atau hanya 9 baris yang terfilter. Atau dapat dikatakan sangat baik.

Dari hasil percobaan menunjukkan bahwa banyak terjemahan kata yang masih menggunakan kata bahasa sumber karena corpus yang digunakan masih kurang lengkap baik kalimat maupun sinonim kata atau frasenya. Sedangkan ke-paralel-an dari *corpus* yang sangat terjaga secara umum menunjukkan hasil yang lebih baik dibandingkan dengan corpus kitab Injil. Tetapi perlu diperhatikan bahwa pembuatan *parallel corpus* yang kurang tepat dapat membuat kesalahan seperti contoh di bawah bahwa frase "apai kin" seharusnya diterjemahkan sebagai "yang mana kah" bukan "manakah lebih" karena menjadi salah artinya, walaupun probabilitas-nya cukup kecil yaitu 0,12. Karena jika ada kalimat "apai kin ajangndu" akan diterjemahkan menjadi "manakah lebih milikmu", dimana seharusnya adalah "yang mana kah punyamu".

*apai kin sukahen , ngatakenca ||| manakah lebih mudah , mengatakan ||| 1 0.0020771 1 0.00598321 2.718*
*apai kin sukahen , ||| manakah lebih mudah , ||| 1 0.0103855 1 0.0179496 2.718*
*apai kin sukahen ||| manakah lebih mudah ||| 1 0.0109428 1 0.0187377 2.718*
***apai kin ||| manakah lebih ||| 1 0.12037 1 0.0187377 2.718***
*apai ||| yang mana ||| 0.5 0.0650253 1 0.111111 2.718*

Demikian juga pada contoh berikut, terjemahan yang benar dari "ateku" atau "ate ku" adalah "saya mau" seperti pada baris ke-

3, tetapi tidak tepat dengan " maunya" seperti pada baris ke-2, sementara itu keduanya mempunyai probabilitas yang sama yakni 0.66 sehingga kemungkinan besar akan menyebabkan kesalahan terjemahan.

```
ateku ||| mau ||| 0.166667 0.214286 0.2 0.333333 2.718 ||| ||| 6 5
ateku ||| maunya ||| 0.666667 0.5 0.4 0.222222 2.718 ||| ||| 3 5
ateku ||| saya mau ||| 0.666667 0.158867 0.4 0.111111 2.718 ||| ||| 3 5
```

### 6.3 Hambatan Karakteristik Bahasa pada Penerapan SMT

Bahasa Indonesia dan Karo mempunyai beberapa karakteristik yang cukup menghambat dalam penerapan SMT, contoh pertama adalah akhiran "ku", "mu", "nya" dalam bahasa Indonesia dan akhiran "ku", "ndu", "ngku", "na", "ta" dalam bahasa Karo.

Dalam pemakaiannya semua akhiran itu dapat ditambahkan pada kata benda untuk menunjukkan kepemilikan terhadap benda tersebut, seperti pada contoh berikut:

Indo: bajuku, sepedamu, kucingnya
[baju saya], [sepeda kamu], [kucing dia]
Karo: bajuku, bajungku, senndu, bajuna, rumahta
[baju saya], [uang kamu], [baju kamu], [rumah kita]

Dalam hal ini, jika dilakukan pendekatan sinonim frase atau kata maka harus dibuat pembuatan sinonim untuk seluruh kemungkinan dari setiap kata benda ditambah dengan akhiran tersebut. Hal ini tidak bisa dihindarkan karena dalam penulisannya akhiran tersebut menempel pada kata benda. Berbeda dengan bahasa Inggris, misalnya, dalam kepemilikan benda seperti: "*my book*", "*your car*", "*our house*", "*his friend*", "*her dress*", dan sebagainya, yang menggunakan kata terpisah atau bukan sebagai akhiran pada kata.

Contoh kedua adalah kendala awalan "ber" pada bahasa Indonesia dan awalan "er" pada bahasa Karo, seperti pada contoh berikut:

Indo: berperang, berkurang, bertambah
[melakukan perang], [makin kurang], [makin tambah]
Karo: erjuma, erturang, erdalan, erguak
[ber ladang], [ber saudara], [ber jalan], [ber bohong]

Contoh ketiga adalah awalan "di" yang diikuti dengan akhiran "i" dalam bahasa Indonesia, dan "i" pada bahasa Karo.

Indo: diperangi, direstui, ditambahi
[diperangi], [diizinkan], [makin tambah]
Karo: dalani, idahi, getuki
[dijalani], [mengunjungi], [dicubiti]

### 6.4 Interpretasi Model

Berdasarkan percobaan penerjemahan bahasa dengan model dari dua *parallel corpus* yang berbeda yaitu corpus dari kitab Injil dan corpus contoh kalimat dan sinonim kata dari nara sumber dengan hasil pengujian diperoleh hasil bahwa nilai rata-rata kualitas terjemahan dengan *parallel corpus* dari kitab Injil lebih baik dibandingkan dengan *parallel corpus* dari nara sumber, yaitu [1,9 | 2,6] berbanding [1,8 | 2,6], seperti dirangkum pada Tabel 3. Tetapi dengan catatan bahwa jumlah kata pada corpus pertama sekitar 9 kali lipat lebih banyak dari corpus kedua, yaitu [86.929 | 97.270] berbanding [10.042 | 9.336], demikian pula sampel pengujian mayoritas diambil dari sampel kalimat kitab Injil sehingga cenderung memberi "keuntungan" pada hasil skor *corpus* pertama.

Tabel 3. Sumber *Parallel Corpus* dan Nilai Skor

| No | Sumber *Parallel Corpus* | Jumlah baris kalimat | Jumlah kata | Nilai Skor *Fluency* | Nilai Skor *Adequacy* |
|---|---|---|---|---|---|
| 1 | Kitab Injil Bahasa Indonesia & bahasa Karo | 4.130 & 4.130 | 86.929 & 97.270 | 1,9 | 2,6 |
| 2 | Kumpulan kalimat dan sinonim kata Bahasa Indonesia & Karo | 5.794 & 5.794 | 10.042 & 9.336 | 1,8 | 2,6 |

Untuk bekerja dengan baik SMT memerlukan jumlah *corpus* yang baik dan banyak. Pada penelitian ini baru digunakan sebagian dari kitab Injil yakni sekitar 90 ribu kata, dari keseluruhan lebih dari 650 ribu kata dalam kitab Injil. Dengan melakukan sejumlah perbaikan pada teks baris/kalimat Alkitab sehingga ke-paralel-an teks bahasa Indonesia dan bahasa Karo lebih "lurus" seperti digambarkan pada Gambar 9 akan menghasilkan *parallel corpus* yang jauh lebih baik.

Demikian juga dari hasil percobaan di atas bahwa dengan membuat sinonim kata atau frase yang baik akan memberikan kontribusi peningkatan kualitas terjemahan yang signifikan, sekalipun dengan jumlah frase/kata yang tidak banyak. Hal ini dapat dilakukan sebagai pendekatan untuk mendapatkan penerjemah SMT yang cukup baik dengan jumlah kalimat yang lebih sedikit.

## 7. Kesimpulan dan Saran

Kesimpulan:

Percobaan penggunaan teknik *Statistical Machine Translation* (SMT) pada penerjemahan dua arah bahasa Indonesia ke bahasa daerah Karo yang menggunakan dua sumber *parallel corpus* menunjukkan hasil pengujian yang masih kurang yaitu skor *fluency* 1,9 dan 1,8 dan *adequacy* 2,6 dari skala 5 atau hanya mendekati kategori tidak pas (*disfluent)*, tetapi hal ini cukup menjanjikan mengingat jumlah kata pada *parallel corpus* yang digunakan masih sedikit dibandingkan dengan jumlah jutaan kata yang digunakan pada penelitian lainnya. Kualitas terjemahan akan meningkat bila jumlah dan kualitas kalimat dalam *parallel corpus* dapat ditingkatkan. Pendekatan dengan cara pembuatan sinonim frase kata yang baik sebagai *corpus* menunjukkan peningkatan skor pengujian terjemahan secara signifikan, walaupun dengan jumlah *corpus* yang tidak banyak.

Penerjemah dengan menggunakan *toolkit* SMT *Moses* ini berhasil diintegrasikan dalam situs web yang interaktif, yang kedepannya setelah perbaikan dan penambahan jumlah kata pada *parallel corpus*, dapat dipublikasikan diinternet untuk digunakan sebagai situs untuk diakses oleh publik untuk pembelajaran bahasa daerah khususnya bahasa Karo, ataupun pada situs web yang sudah ada untuk meningkatkan kunjungan *surfer* ke suatu situs web atau disebut *web traffic*.

Saran:

Dengan lisensi LGPL *open source* dari *toolkit* SMT yang digunakan, seperti dijelaskan dalam penelitian ini, maka sangat terbuka peluang untuk pengembangan lebih jauh dalam hal penelitian alat penerjemah bahasa-bahasa daerah di Indonesia, yang sekaligus berfungsi sebagai cara untuk melestarikan kekayaan bahasa di Indonesia supaya tidak cepat punah.

## Daftar Pustaka

[1] Lewis, M. Paul (ed.), "*Ethnologue: Languages of the World*", Sixteenth edition. Dallas, Tex.: SIL International, 2009.

[2] Yurnaldi, "169 Bahasa Daerah Terancam Punah", http://nasional.kompas.com/read/2008/08/11/21544654/169.bahasa.daerah.terancam.punah, (Diakses 14 Januari 2012)

[3] Maffi, Luisa, "Language: A Resource for Nature", *The UNESCO Journal on the Environment and National Resources Research*, 1998.

[4] Brown, Peter F., et.al, "*A Statistical Approach to Language Translation*", IBM T.J. Watson Research Center, Yorktown Heights, 1990.

[5] Ramis, Adria de Gispert, "Introducing Linguistic Knowledge into Statistical Machine Translation", *PhD Thesis Dessertation*, Universitat Politecnica De Catalunya, Spain, 2006.

[6] Brown, Peter F., et.al, "*The Mathematics of Statistical Machine Translation*", IBM T.J. Watson Research Center, Yorktown Heights, 1993.

[7] Xiong, Deyi, et.al, "Enhancing Language Models in Statistical Machine Translation with Backward N-grams and Mutual Information Triggers", *Proceedings of the 49th Annual Meeting*

*of the Association for Computational Linguistics*, pages 1288–1297, Portland, Oregon, 2011.

[8] Monz, Christof, "Statistical Machine Translation with Local Language Models", Informatics Institute, University of Amsterdam for *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 869–879*, Edinburgh, Scotland, UK, July 27–31, 2011.

[9] Koehn, Philipp, "*MOSES – Statistical Machine Translation System User Manual and Code Guide*", University of Edinburgh, 2012.

[10] Koehn, Philipp, et.al, "Open Source Toolkit for Statistical Machine Translation: Factored Translation Models and Confusion Network Decoding", *Final Report of the 2006 Language Engineering Workshop, Johns Hopkins University Center for Speech and Language Processing*, 2007.

[11] Knight, Kevin and Koehn, Philipp, "*What's New in Statistical Machine Translation*", Information Sciences Institute, University of Southern California, 2011.

[12] Koehn, Philipp, "*European Parliament Proceedings Parallel Corpus 1996-2011*", http://www.statmt.org/europarl/, (Diakses 14 Januari 2012)

[13] Tian, Liang, et.al, "*Word Alignment Using GIZA++ on Windows*", Department of Computer and Information Science University of Macau, Macau, S.A.R, China, 2009.