UNIVERSITAS INDONESIA

CROSS LINGUAL SENSE TRANSFERING UNTUK MEMBANGUN *SENSE TAGGED CORPUS* DAN *WORD SENSE DISAMBIGUATION* (WSD) BAHASA INDONESIA

SKRIPSI

ADITYA RAMA
1306397854

FAKULTAS ILMU KOMPUTER
PROGRAM STUDI ILMU KOMPUTER
DEPOK
JUNI 2017

UNIVERSITAS INDONESIA

CROSS LINGUAL SENSE TRANSFERING UNTUK MEMBANGUN *SENSE TAGGED CORPUS* DAN *WORD SENSE DISAMBIGUATION* (WSD) BAHASA INDONESIA

SKRIPSI
Diajukan sebagai salah satu syarat untuk memperoleh gelar
Sarjana Ilmu Komputer

ADITYA RAMA
1306397854

FAKULTAS ILMU KOMPUTER
PROGRAM STUDI ILMU KOMPUTER
DEPOK
JUNI 2017

HALAMAN PERNYATAAN ORISINALITAS

Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.

Nama	:	**Aditya Rama**
NPM	:	**1306397854**
Tanda Tangan	:	
Tanggal	:	**5 Juni 2017**

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh	:	
Nama	:	Aditya Rama
NPM	:	1306397854
Program Studi	:	Ilmu Komputer
Judul Skripsi	:	*Cross Lingual Sense Transfering* Untuk Membangun *Sense Tagged Corpus* dan *Word Sense Disambiguation* (WSD) Bahasa Indonesia

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Ilmu Komputer pada Program Studi Ilmu Komputer, Fakultas Ilmu Komputer, Universitas Indonesia.

DEWAN PENGUJI

Pembimbing	:	Mirna Adriani, Dra, Ph.D.	(	)
Pembimbing	:	Rahmad Mahendra, S.Kom., M.Sc.	(	)
Penguji 1	:		(	)
Penguji 2	:		(	)

Ditetapkan di	:	Depok
Tanggal	:	Juni 2017

KATA PENGANTAR

Segala puji bagi Allah SWT atas segala rahmat dan karunia yang telah diberikan, sehingga laporan penelitian ini dapat diselesaikan untuk memenuhi salah satu syarat menyelesaikan pendidikan program Sarjana Ilmu Komputer Universitas Indonesia. Penulis ingin menyampaikan terima kasih untuk pihak-pihak yang sudah secara langsung maupun tidak langsung membantu jalannya penelitian ini, diantaranya:

1. Keluarga penulis baik itu Ibu, Ayah, dan juga Kakak yang telah memberikan doa dan segala dukungan yang dibutuhkan selama jalannya penelitian.
2. Dra. Mirna Adriani, Ph.D dan Rahmad Mahendra, S.Kom., M.Sc. sebagai dosen pembimbing yang telah memberikan arahan, bimbingan, dan motivasi untuk menyelesaikan skripsi ini.
3. Nadiarani yang selalu membantu dalam memberikan semangat dan juga menuntut hasil penulisan secara berkala agar mendorong keinginan untuk menulis.
4. Jodi Prayogo, Hartico, Ilham Kurniawan, dan Firza Pratama sebagai pemberi masukannya pada pembuatan panduan anotasi dan juga proses evaluasi *word alignment*.
5. Teman-teman angkatan 2013 (Angklung) yang saling memberi dukungan dan doa untuk kelancaran setiap orang di dalamnya.

Depok, Mei 2017

Aditya Rama

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama	:	Aditya Rama
NPM	:	1306397854
Program Studi	:	Ilmu Komputer
Fakultas	:	Ilmu Komputer
Jenis Karya	:	Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (Non-exclusive Royalty Free Right)** atas karya ilmiah saya yang berjudul:

Cross Lingual Sense Transfering Untuk Membangun *Sense Tagged Corpus* dan *Word Sense Disambiguation* (WSD) Bahasa Indonesia

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia-/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyatan ini saya buat dengan sebenarnya.

Dibuat di : Depok
Pada tanggal : 5 Juni 2017

Yang menyatakan

(Aditya Rama)

ABSTRAK

Nama	:	Aditya Rama
Program Studi	:	Ilmu Komputer
Judul	:	*Cross Lingual Sense Transfering* Untuk Membangun *Sense Tagged Corpus* dan *Word Sense Disambiguation* (WSD) Bahasa Indonesia

Cross Lingual Word Sense Disambiguation (CLWSD) merupakan salah satu pendekatan untuk menyelesaikan permasalahan disambiguasi makna kata di bidang NLP. Pendekatan ini memanfaatkan sebuah konsep dimana suatu kata dapat diterjemahkan menjadi beberapa kata yang berbeda tergantung dengan konteks dimana kata tersebut muncul. Keterbatasan data berupa *sense tagged corpus* menjadi salah satu permasalahan yang menghambat penelitian WSD di bahasa Indonesia ini. Pada penelitian kali ini, pendekatan CLWSD akan digunakan untuk *transfer sense* dari *sense tagged corpus* bahasa Inggris ke bahasa Indonesia dengan memanfaatkan korpus paralel dwibahasa. Hasil dari penelitian ini merupakan *sense tagged corpus* dalam bahasa Indonesia yang kemudian juga akan dicoba dalam sistem WSD yang dibuat sendiri. Pada penelitian ini, *sense transfering* berhasil menghasilkan *sense tagged corpus* dalam bahasa Indonesia. Selain *sense tagged corpus* tersebut, sistem WSD yang dibangun juga memiliki performa diatas baseline pembanding pada *target word* dan fitur tertentu yang diberikan.

Kata Kunci:
Cross Lingual, Word Sense Disambiguation

ABSTRACT

Name : Aditya Rama
Program : Computer Science

Title : Cross Language Sense Transfering To Build Sense Tagged Corpus and Word Sense Disambiguation for Bahasa Indonesia

Cross Lingual Word Sense Disambiguation (*CLSWD*) is one among the methods in solving word sense disambiguation problem in NLP field. This approach utilize a concept that a word can translated to many words depend on where that word appear. Limitation of data (*sense tagged corpus* in Indonesian language) become one problem that hold the development of research in Indonesian WSD. In this research, CLWSD approach will be used to transfer sense from english sense tagged corpus into Indonesian by using parallel corpora. Result of this research is a sense tagged corpus in Indonesian language that will be tested by our implemented WSD system. Based on the result of the experiment, we could see that Indonesian sense tagged corpora has been built by the proposed method. Beside the sense tagged corpora, the WSD system itself also having performance above the baseline on some given target words and features.

Keywords:
Cross Lingual, Word Sense Disambiguation

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR KODE

BAB 1
PENDAHULUAN

Bab ini membahas mengenai latar belakang penelitian, perumusan masalah, tujuan dan manfaat penelitian, ruang lingkup penelitian, metodologi penelitian, serta sistematika penulisan.

1.1 Latar Belakang

Word Sense Disambiguation (WSD) merupakan salah satu tugas untuk menentukan makna terbaik dari sebuah kata. Sebuah kata sendiri dapat memiliki beberapa makna dan bergantung pada konteks dimana kata tersebut muncul. Penentuan makna kata yang paling tepat ini secara tidak langsung dapat membantu beberapa *task Natural Language Processing*(NLP) ataupun *Information Retrieval* (IR) lainnya seperti misalnya *machine translation* (MT). Salah satu contoh dimana WSD dapat membantu dalam sistem IR adalah pada *task question answering* berikut.

```
Q: Apakah orang Indonesia sarapan apel di pagi hari?
A: Ya, orang indonesia sering melakukan apel di pagi hari setelah
    mereka sarapan dan pergi ke kantor
```

Pada contoh kasus *question answering* diatas, dapat terjadi jawaban dari *query* yang diberikan tidak cocok dengan pertanyaan karena sistem tidak mengerti bahwa "apel" yang dimaksud pada kalimat pertanyaan adalah apel (buah), bukan merupakan apel (upacara). Sistem WSD yang dapat melakukan disambiguasi makna kata dapat membantu sistem QA diatas untuk dapat mengetahui bahwa "apel" dalam pertanyaan memiliki makna sebagai buah.

Membangun sistem WSD biasanya dilakukan dengan beberapa pendekatan *machine learning* seperti *supervised learning*, *semi supervised*, ataupun *unsupervised*. Pendekatan *supervised machine learning* yang dapat digunakan untuk membangun sistem WSD membutuhkan data yang tidak sedikit. Data yang dibutuhkan untuk sistem ini dapat berupa *sense-tagged corpus* dimana isinya adalah kata-kata yang sudah mempunyai kelas makna kata yang tepat. Kebutuhan akan data yang relatif besar tersebut merupakan kendala yang ada pada bahasa-bahasa tertentu. Bahasa Inggris sebagai salah satu bahasa internasional mempunyai data yang cukup banyak untuk membangun sistem dengan *supervised learning*. Namun demikian, bahasa Indonesia sendiri termasuk dalam *under resource language*

dimana data yang dapat dimanfaatkan untuk sistem WSD masih terbatas. Belum adanya data seperti *sense-tagged corpus* untuk membangun sistem WSD bahasa Indonesia merupakan salah satu permasalahan yang dihadapi jika dibandingkan dengan bahasa Inggris.

Makna kata yang diberikan oleh sistem WSD pada kata-kata pada umumnya berasal dari definisi pada Wordnet. Namun demikian, membangun Wordnet secara manual untuk memenuhi kebutuhannya sebagai inventaris makna kata membutuhkan waktu dan dana yang relatif tidak sedikit. Terdapat beberapa Wordnet bahasa Indonesia yang dapat digunakan, diantaranya adalah Wordnet Bahasa (bahasa.cs.ui.ac.id) dan Wordnet Bahasa yang berasal dari proyek Nanyang Technological University (NTU). Wordnet dari Bahasa Fasilkom UI masih memiliki sedikit *sets of synonyms* (synsets) yang mana berarti banyak kata-kata yang tidak tersedia di dalamnya. Wordnet bahasa dari proyek NTU sendiri masih mengandung makna kata yang tidak sesuai. Oleh karena masalah tersebut, pada penelitian ini Wordnet yang digunakan adalah Wordnet bahasa Inggris buatan Princeton yang sudah sering digunakan untuk *event* Semantic Evaluation atau Sense Evaluation. Penggunaan makna kata dari Wordnet bahasa Inggris tersebut nantinya akan dipindahkan ke kata dalam bahasa Indonesia yang bersesuaian dengan memanfaatkan sifat paralel dari korpus identik hasil penelitian (Larasati, 2012).

Pendekatan *cross lingual sense transfering* dengan bahasa Inggris sebagai pasangan korpus diharapkan dapat memperkaya *resource*(data) yang masih kurang pada bahasa Indonesia.

1.2 Perumusan Masalah

Beberapa pertanyaan yang menjadi rumusan masalah dalam penelitian ini yaitu:

1. Bagaimana cara membangun *sense tagged corpus* Bahasa Indonesia dengan pendekatan *cross lingual* dari paralel korpus?

2. Seberapa baik performa WSD dari *sense tagged corpus* pada tahap pertama?

1.3 Tujuan dan Manfaat Penelitian

Tujuan dari penelitian ini adalah menghasilkan *sense tagged corpus* dalam bahasa Indonesia, dan menghitung seberapa baik performa *wsd system* bahasa Indonesia yang dibuat.

1.4 Ruang Lingkup Penelitian

Penelitian berfokus pada pemindahan *sense* dari korpus paralel berbahasa Inggris ke Indonesia, dan melakukan *WSD task* dari hasil *sense transfering* tersebut.

1.5 Metodologi Penelitian

Ada lima tahapan yang dilakukan pada penelitian ini. Penjelasan dari tiap tahapan adalah sebagai berikut.

1. Studi Literatur
   Tahap ini berfokus pada pencarian informasi mengenai *WSD system* baik secara umum maupun teknik yang digunakan, dan juga *task* lain yang berkaitan dengan *WSD* seperti *word sense induction* (*WSI*).

2. Perumusan Masalah
   Masalah-masalah yang ada dalam penelitian nantinya dianalisis penyelesaiannya pada tahap ini.

3. Rancangan Penelitian
   Proses yang melibatkan seluruh penelitian untuk menyelesaikan permasalahan yang ada.

4. Implementasi
   Tahap ini merupakan implementasi dari rancangan yang sudah dibuat untuk memecahkan permasalahan yang ada.

5. Analisis dan Kesimpulan
   Hasil percobaan dianalisis untuk mendapatkan gambaran seberapa baik performa dari sistem yang dibuat.

1.6 Sistematika Penulisan

Sistematika penulisan yang ada dalam laporan penelitian ini sebagai berikut:

- Bab 1 PENDAHULUAN
  Bab ini akan menjelaskan mengenai latar belakang, perumusan masalah, tujuan penelitian, tahapan penelitian, ruang lingkup, metodologi, dan sistematika penulisan dari penelitian ini.

- Bab 2 TINJAUAN PUSTAKA
  Bab ini akan menjelaskan mengenai konsep dan teori yang relevan dari hasil studi literatur yang telah dilakukan. Teori-teori yang dijelaskan meliputi *Word sense disambiguation*, *Word sense induction*, dan beberapa hal lain yang dibutuhkan pada penelitian.

- Bab 3 RANCANGAN PENELITIAN
  Bab ini akan membahas perihal rancangan dari proses penelitian yang meliputi *sense tagging* korpus Inggris, *word alignment* Indonesia-Inggris, *sense transfering*, dan sistem WSD bahasa Indonesia.

- Bab 4 IMPLEMENTASI
  Pada bab ini akan menjelaskan mengenai implementasi dari rancangan sistem yang dibuat.

- Bab 5 HASIL DAN ANALISIS
  Pada bab ini, dijelaskan mengenai hasil penelitian beserta evaluasi dan analisisnya.

- Bab 6 PENUTUP
  Kesimpulan dan saran dari hasil dan pelaksanaan penelitian akan dijelaskan pada bab ini.

BAB 2
TINJAUAN PUSTAKA

Bab ini membahas mengenai studi literatur yang digunakan selama penelitian. Studi literatur ini menjelaskan tentang hal-hal mendasar yang dibutuhkan dalam penelitian.

2.1 Lexical *Semantic*

Kata dalam suatu bahasa memiliki banyak fitur atau informasi yang terkandung di dalamnya. Informasi pada kata tersebut meliputi contoh-contoh seperti ortografi, POS Tag, dan juga makna kata. Informasi yang ada tersebut dapat dipergunakan sebagai fitur untuk merepresentasikan suatu kata.

2.1.1 *Word Feature*

Ortografi meliputi permasalahan dalam suatu kata pada bahasa mengenai ejaan, kapitalisasi, pemenggalan kata, dan lain-lain. Pada bentuk kata (morfologi), Lemma merupakan bentuk kata "dasar" seperti misalnya "*run*", *"ran"*, dan "*runs*" memiliki lemma yang sama yaitu "run". POS Tag atau Part of Speech Tag merupakan "kategori" dari suatu kata yang didefinisikan untuk membedakan kelas-kelas yang ada seperti misalnya *proper noun* (nama orang, organisasi, nama tempat, dan lain-lain), *verb* (kata kerja), *adjective* (kata sifat seperti "marah", "senang", dan lain-lain), *adverbial*, dan semacamnya. *Tool* yang digunakan untuk melakukan POS Tagging pada penelitian ini adalah *tool* dari Stanford. Makna kata menyimpan informasi dari arti sebuah kata, dan satu kata bisa memiliki lebih dari satu makna. Selain informasi dalam suatu kata, terdapat relasi antara satu kata dengan kata lain seperti misalnya sinonim, homonim, polisemi, dan lain-lain. Berdasarkan informasi (Sheeba et al., 2013), sinonim merupakan relasi pada kata-kata yang berbeda namun memiliki makna yang serupa, dan homonim merupakan kata dengan ejaan atau penulisan yang sama namun memiliki makna yang berbeda. Mirip dengan homonim, polisemi merupakan suatu kata yang memiliki makna lebih dari satu. Contoh dari relasi sinonim adalah antara kata "bohong" dengan "dusta", "perspektif" dengan "sudut pandang", dan kata-kata dengan makna mirip lainnya. Contoh dari kata polisemi adalah pada kata "darah" dalam kalimat "Tangan Ani mengeluarkan darah setelah tertusuk jarum" dengan "Ani baru menyadari

bahwa dia dan Rina memiliki hubungan darah". Relasi homonim antar kata dapat dilihat misalnya pada kata "rapat" yang dapat berarti "pertemuan" ataupun juga "berdekatan", sebenarnya kata ini merupakan dua buah kata yang berbeda(sehingga maknanya berbeda) hanya saja penulisan dan pengejaannya sama.

Terdapat beberapa cara untuk merepresentasikan sebuah kata sebagai fitur. Salah satu cara yang sederhana adalah dengan merepresentasikan kata sebagai *one hot vector*. Pada model ini, setiap kata di dalam sebuah korpus diberikan nomor indeks untuk membangun vektor yang mewakili keberadaan kata tersebut. Jika terdapat sebuah kata yang muncul pada konteks yang ingin direpresentasikan, indeks vektor yang sama dengan indeks kata tersebut akan bernilai 1. Bila terdapat sebuah korpus dengan jumlah kata unik berjumlah 4 dengan kata-kata "Ani", "marah", "kemarin", dan "malam" (sebuah vektor dengan pangjang empat). Representasi *one hot vector* untuk kalimat "Ani marah" dapat ditulis dengan vektor [1,1,0,0].

Representasi *one hot vector* akan mempunyai panjang vektor yang besar jika korpus mempunyai jumlah kata unik yang besar. Terdapat bentuk representasi lain untuk membentuk vektor dari kata, salah satunya adalah dengan *word embedding*. *Word Embedding* menggunakan representasi bilangan *real* pada vektor untuk merepresentasikan sebuah kata berdasarkan hasil *training* dengan suatu korpus. Contoh representasi dari *word embedding* pada suatu kata "makan" adalah vektor [0.6, -0.3, ..., 0.5] (misalnya). Vektor dari hasil *word embedding* mempunyai karakteristik dimana jarak antara dua buah vektor dari kata yang mirip secara semantik bernilai kecil(dekat). Bila misalkan pada data *training* untuk *word embedding* terdapat banyak kalimat-kalimat berbentuk "... makan Y ..." dimana Y adalah sebuah objek berupa makanan. Maka kata-kata yang mewakili Y seperti misalnya "burger", "apel", "steak", dan lain-lain, akan memiliki vektor yang mirip dan secara implisit dapat saling menggantikan untuk menempati posisi Y tersebut. Berdasarkan keterdekatan vektor tersebut, *word embedding* mampu untuk menangkap semantik dari kata-kata yang ada pada korpus.

Salah satu model *word embedding* yang dapat digunakan adalah Word2Vec (Mikolov et al., 2013). Pada penelitian ini Word2Vec akan digunakan untuk memanfaatkan vektor *word embedding* sebagai salah satu fitur percobaan skenario yang ada.

2.1.2 Wordnet

Wordnet merupakan *online lexical database* yang didesain dan untuk digunakan suatu program (Miller, 1995). Salah satu Wordnet untuk bahasa Inggris yang sudah

berkembang dan digunakan dalam berbagai penelitian adalah Wordnet yang dikembangkan Princeton. Representasi yang digunakan Wordnet adalah *set of synonyms* (synset) sebagai kumpulan dari kata-kata yang memiliki kemiripan makna. Selain menyimpan kata-kata, Wordnet juga menyimpan makna dari setiap kata-kata tersebut. Informasi makna kata yang ada di dalam Wordnet sering digunakan untuk sistem WSD suatu bahasa sebagai acuan (*sense inventory*). Makna kata pada Wordnet disimpan dengan *uniq identifier* berupa *sense key* yang memiliki format "**lemma%key**". Salah satu contoh *sense key* adalah "home%1:06:00::" yang mana memiliki makna "*Housing that someone is living in*". Representasi *synset* dari Wordnet beserta relasi-relasinya tersebut dapat dimanfaatkan untuk disambiguasi makna suatu kata.

2.2 Word Sense Disambiguation

Word Sense Disambiguation merupakan salah satu penelitian di bidang NLP yang bertujuan untuk menentukan makna yang paling tepat dari suatu kata berdasarkan konteks kata tersebut ditemukan. Sebagaimana kata dalam suatu bahasa bisa memiliki makna lebih dari satu (polisemi), *task* ini akan menentukan makna kata mana yang paling tepat.

Penentuan makna kalimat dilakukan dengan pemberian informasi berupa kata yang menjadi *target* dan konteks berupa kalimat. Contoh proses disambiguasi yang dilakukan untuk kata **cokelat**:

```
K1: Roni memakan cokelat yang diberikan ibunya
K2: Walaupun mobil cokelat itu mahal, dia sangat ingin membelinya
```

Pada kalimat pertama (K1), **cokelat** yang dimaksud memiliki makna sebagai makanan yang terbuat dari buah *cokelat*. Sementara itu, Kata **cokelat** pada kalimat kedua (K2) memiliki makna yang berbeda, dimana kata tersebut merupakan satu keterangan warna. Penentuan makna yang tepat dapat dilakukan dengan bantuan informasi konteks dari kalimat dimana kata tersebut muncul. Pada K1, kata **memakan** memberikan informasi bahwa *cokelat* yang dimaksud adalah objek yang bisa dimakan. Kata yang memberikan informasi pada kalimat kedua adalah kata **berwarna** yang secara eksplisit menerangkan bahwa **cokelat** yang dimaksud adalah warna. Namun demikian, konteks maupun informasi yang bisa diambil dari kalimat tidak selalu eksplisit. Pada contoh kalimat seperti "Pohon cokelat tua di belakang rumahku sangat besar", cokelat yang dimaksud bisa bermakna "buah cokelat yang sudah tua" atau "berwarna cokelat tua".

Penentuan makna kata yang tepat oleh sistem WSD ditentukan berdasarkan

konteks dari kata tersebut berada. Walaupun satu kata dapat memiliki beberapa makna, terdapat kecil kemungkinan bahwa kata yang sama digunakan dalam satu *discourse* (konten dari sebuah percakapan atau bahasan) untuk menyatakan makna yang berbeda sebagaimana "*one sense per discourse*" (Gale et al., 1992).

Pada umumnya, sistem WSD yang dibangun dapat menggunakan pendekatan *machine learning* baik itu *supervised*, *semi-supervised*, maupun *unsupervised*. Pendekatan yang dipilih biasanya bergantung pada data maupun *resource* yang dimiliki pada suatu bahasa. Jika terdapat data/*resource* yang memadai, sistem dengan pendekatan *supervised* biasanya dilakukan untuk mendapatkan akurasi yang optimal. Namun demikian, untuk bahasa-bahasa dengan data/*resource* yang kurang memadai, pendekatan *semi-supervised* atau bahkan *unsupervised* dapat dimanfaatkan untuk mendapatkan makna kata secara otomatis. Terdapat dua buah granularitas dalam sistem WSD yaitu *coarse-grain* dan *fine-grain*. Perbedaan utama yang membedakan adalah *fine-grain* memiliki *sense inventory* yang lebih detail dan dengan kemiripan makna antar kata yang lebih rinci dibandingkan dengan *coarse-grain*. Pada sistem WSD, fitur yang umum digunakan beberapa diantaranya adalah konteks kata baik dengan pendekatan Collocation dan Co-occurrence, POS Tag kata, vektor *word embedding* kata. Fitur Co-occurrence lebih berfokus pada kemunculan kata pada suatu konteks baik itu dokumen, paragraf, atau kalimat, sementara Collocation lebih berfokus pada kata yang sering muncul secara bersama-sama dan kecil kemungkinannya untuk muncul bersama secara kebetulan.

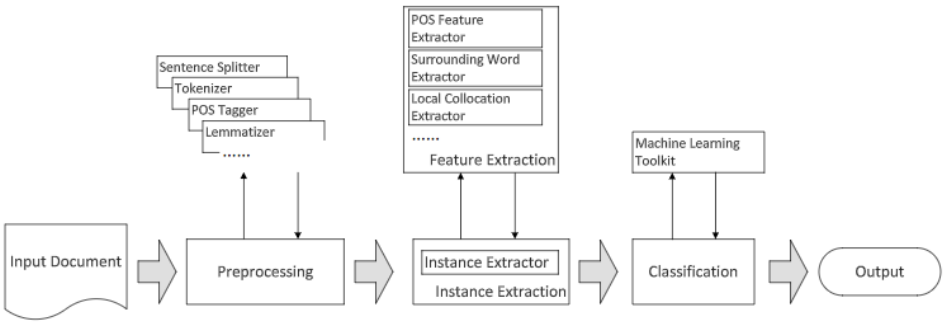
2.2.1 WSD Bahasa Indonesia

Salah satu penelitian WSD Bahasa Indonesia dilakukan dengan pendekatan *cross lingual* (Septiantri, 2013). Pada penelitian tersebut, proses diawali dengan mendapatkan makna kata dengan memanfaatkan korpus paralel dwibahasa. Proses ekstraksi makna kata tersebut akan dijelaskan pada sub-bab *cross language* dalam pembahasan *Word Sense Induction*. Setelah makna kata didapatkan, kata-kata yang memilki makna lebih dari satu diambil untuk dijadikan *sample* untuk mengevaluasi sistem WSD yang dibangun. Terdapat beberapa *classifier* yang digunakan untuk dijadikan perbandingan yaitu Naive Bayes, Random Forest, dan SVM.

2.2.2 WSD Bahasa Inggris

Terdapat berbagai penelitian yang sudah dilakukan untuk melakukan disambiguasi kata pada Bahasa Inggris. Namun demikian, terdapat salah satu sistem WSD yang sudah dipublikasikan dan siap digunakan untuk melakukan WSD dengan *input*

berupa *free text* dalam Bahasa Inggris. Sistem WSD tersebut adalah "It Makes Sense" (IMS) yang dibuat oleh Zhi Zhong dan Hwee Tou Ng (Zhong dan Ng, 2010). Sistem dibangun menggunakan pendekatan *supervised learning* yang dapat digunakan untuk semua kata bahasa Inggris. Pada dasarnya, *classifier* yang dipilih untuk *task* ini adalah *support vector machine* (SVM). Arsitektur yang dibangun pada IMS dapat dilihat pada gambar berikut:



Gambar 2.1: Arsitektur IMS

Proses *pre-processing* pada IMS dilakukan dengan empat tahapan:

1. Mendeteksi batasan kalimat dengan *sentence splitter*
2. Tokenisasi dengan *tokenizer*
3. POS Tagging untuk semua token
4. Mengubah token menjadi lemma dengan *lemmatizer*

Ekstraksi fitur dilakukan dengan mengombinasikan:

1. POS Tag dari tiga buah kata di kiri dan kanan *target word*, serta kata itu sendiri.
2. Kata-kata sekitar pada konteks kalimat ataupun kalimat tetangganya. Kata-kata yang terkandung di dalam *stopwords* dan memiliki simbol atau angka dibuang dari kalimat tersebut. Kata-kata yang tersisa tersebut kemudian diubah menjadi bentuk kata dasarnya dalam huruf kecil.
3. *Local Collocation* dengan 11 buah *collocation* baik itu sebelum *target word* maupun setelahnya.

Pengujian seberapa baik performa IMS dalam melakukan WSD *task* mendapatkan hasil:

	SensEval-2	SensEval-3	SemEval-2007	
	Fine-grained	Fine-grained	Fine-grained	Coarse-grained
IMS	68.2%	**67.6%**	58.3%	**82.6%**

Gambar 2.2: Performa IMS (Zhong dan Ng, 2010)

IMS digunakan pada penelitian ini karena sistem WSD ini sudah dapat digunakan untuk *tagging* makna kata dengan input berupa *free text*.

2.3 Word Sense Induction

Word Sense Induction (WSI) adalah sebuah *task* yang mempunyai fungsi utama untuk mendapatkan makna kata dari sebuah korpus atau teks yang belum dianotasi secara otomatis. WSI dapat dilakukan jika penelitian WSD yang ingin dilakukan tidak mempunyai cukup *resource* seperti misalnya Wordnet ataupun *sense tagged corpus* yang *reliable*. *Reliable* pada hal ini dapat berarti kurangnya jumlah *sense* pada data, ataupun masih sedikitnya jumlah kata-kata yang ada pada Wordnet. WSI secara tidak langsung dapat memperbanyak data yang dapat digunakan jika sistem WSD yang ingin dibangun membutuhkan data *training* yang tidak sedikit. Terdapat berbagai macam pendekatan dalam melakukan *sense induction*, diantaranya adalah dengan melakukan *clustering* kata (Denkowski, 2009), ataupun menggunakan pendekatan *cross language*.

2.3.1 Pendekatan *Clustering*

Dua kata dianggap dekat secara semantik jika memiliki *co-occurrence* dengan kata-kata tetangganya yang sama (Nasiruddin, 2013). Konsep tersebut mendasari cara WSI mendapatkan *sense* kata secara implisit berdasarkan hasil *cluster* yang terbentuk dari data atau teks mentah (teks yang tidak dianotasi).

Penarikan makna secara implisit dapat dicontohkan pada beberapa kalimat rujukan berikut (Denkowski, 2009):

1. A bottle of tezgüno is on the table.
2. Everyone likes tezgüno.
3. Tezgüno makes you drunk.
4. We make tezgüno out of corn.

Walaupun belum terdapat informasi eksplisit makna dari tezgüno, dapat disimpulkan bahwa tezgüno mengacu pada minuman beralkohol yang memabukkan. Penarikan kesimpulan ini didapatkan dari kemunculan kata tersebut dengan kata lain pada konteks yang sama.

Pada pendekatan *clustering* ini, makna kata bisa didapatkan secara implisit dari hasil *cluster* yang terbentuk, namun demikian pelabelan yang dilakukan untuk menentukan apa yang direpresentasikan *cluster* tersebut merupakan sebuah *task* tersendiri.

Salah satu cara *clustering* yang dijelaskan (Pantel dan Lin, 2002) adalah dengan mencoba beberapa algoritma seperti K-Means, Buckshot, CBC, Unicon, Bisecting K-Means, dan Average Link untuk fitur yang ditentukan. Fitur yang digunakan adalah vektor dari kata yang didapatkan dengan menghitung *discounted pointwise mutual information* dari kata tersebut pada konteks kata itu muncul. Perhitungan *similarity* antara dua buah kata dihitung dengan menghitung jarak vektor kedua kata tersebtu.

Pada pendekatan *clustering*, hasil dari *cluster* yang didapat masih harus diberikan label dan makna kata tidak bisa secara eksplisit diambil dari hasil tersebut. Proses *topic modeling* masih diperlukan untuk dapat menentukan "topik" apa yang menggambarkan atau merepresentasikan *cluster* tersebut.

2.3.2 Pendekatan *Cross Language*

Selain pendekatan *clustering*, WSI juga dapat memanfaatkan fitur dimana satu kata dari suatu bahasa, dapat diterjemahkan menjadi beberapa kata di bahasa lain. Contoh kasus tersebut dapat dilihat pada kata "halaman" berikut:

```
(K1-Indonesia): Aku membaca 10 halaman buku Harry Potter
(K1-English): I read 10 pages of Harry Potter book
(K2-Indonesia): Ani tinggal di rumah dengan halaman yang sangat
    luas
(K2-English): Ani lives in a house with very large yard
```

Berdasarkan kedua pasangan kalimat tersebut, kata **halaman** dalam bahasa Indonesia dapat diterjemahkan menjadi dua buah kata dalam bahasa Inggris, yaitu *page* ataupun *yard*. Hal ini menunjukan bahwa terjemahan dari suatu kata bergantung pada konteks dimana kata tersebut muncul.

Salah satu penelitian yang juga menggunakan pendekatan *Cross-Lingual* WSI adalah penelitian (Septiantri, 2013) yang memanfaatkan korpus paralel untuk menentukan makna yang tepat dari suatu kata berdasarkan makna tersebut dalam Bahasa Indonesia dan Bahasa Inggris. Proses yang digunakan untuk menentukan

makna kata secara *cross lingual* pada penelitian tersebut dilakukan dengan tahap-tahap:

1. Lakukan pemasangan kata-kata antara kedua korpus (Bahasa Inggris - Indonesia) dengan Giza++

2. Untuk kata-kata pada setiap pasangan tersebut, berikan *tag* makna kata yang mungkin dari Wordnet NTU. Contoh dari proses ini misalnya pada pasangan kata "halaman" dengan "*page*". Berikan *sense id* yang mungkin untuk kata "halaman" dari Wordnet NTU misalnya sense-id-1, sense-id-2, dan sense-id-3. Lakukan hal yang sama pada kata "*page*", misalnya diberikan *sense id* sense-id-5, sense-id-7, dan sense-id-2.

3. Berdasarkan hasil *sense id* dari proses sebelumnya, ambil *sense id* yang beririsan untuk dijadikan *tag* dari *sense* kata tersebut. Pada contoh kata "halaman" dan "*page*" tadi, maka label/*tag* yang diberikan pada kata "halaman" adalah *sense id* sense-id-2.

4. Lakukan proses pelabelan *tagging* makna kata tersebut untuk pasangan kata lainnya.

5. Suatu kata yang memiliki *sense id* lebih dari satu kemudian dinyatakan sebagai kata yang ambigu.

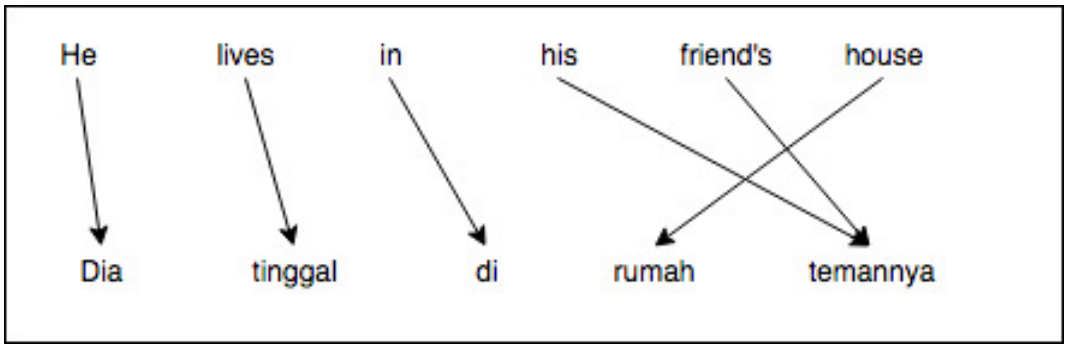
2.4 Korpus Paralel dan *Comparable*

Terdapat dua macam korpus bilingual yang dapat dimanfaatkan untuk pemanfaatan *cross language* WSD yaitu korpus paralel dan *comparable*. Perbedaan utama terhadap kedua buah korpus berada pada seberapa identik kedua buah konteks yang dimilikinya. Korpus paralel memiliki kalimat dan kata-kata yang serupa antara dua buah pasangan di masing-masing korpus. Hal ini dapat dicontohkan misalnya dengan kalimat satu pada korpus bahasa Indonesia "Aku makan" dengan "I eat" pada korpus bahasa Inggris. Salah satu contoh korpus paralel adalah Catholic Bible yang dinamakan sebagai bitext (Rudnick, 2011) yang merupakan kitab katolik dalam bahasa yang berbeda. Berbeda dengan korpus paralel, *comparable* berarti kedua kalimat atau *instance* yang berpasangan hanya sebatas mirip/sama dalam suatu kategori kriteria tertentu. Dengan adanya korpus paralel dan *comparable* tersebut, dibutuhkan juga alat untuk menyelaraskan (*aligning*) konten pada kedua korpus tersebut. *Alignment* yang dapat dilakukan memiliki beberapa tingkatan mulai dari *scope* yang besar sampai kecil. *Scope* besar tersebut meliputi *alignment*

dokumen yang mana fungsinya adalah menyelaraskan antar dokumen yang konten atau kriterianya sama. Tingkatan yang lebih kecil berikutnya yaitu kalimat dimana *alignment* dilakukan pada *level* kalimat (pasangan kalimat yang makna atau kriterianya sama). *Alignment* dengan tingkatan yang lebih spesifik lagi adalah kata (*word alignment*), dimana hasil yang didapat dari proses ini adalah pasangan kata pada kedua korpus dwibahasa yang selaras. Korpus paralel maupun *comparable* ini dapat dimanfaatkan untuk sistem terkait *cross language information retrieval* (CLIR), *machine translation* (MT), CLWSD, dan lain-lain.

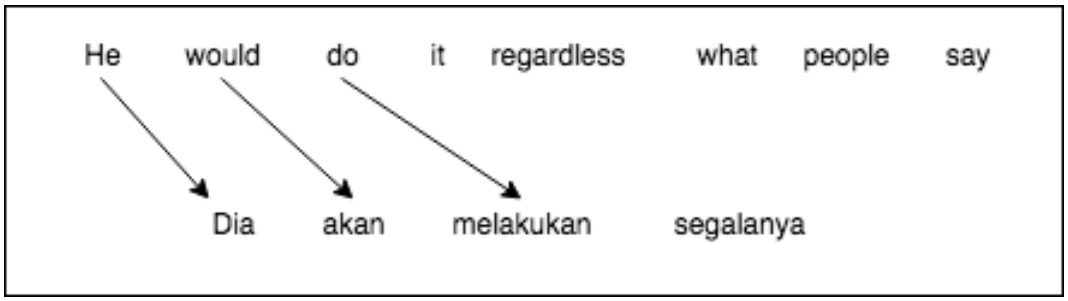
2.5 *Word Alignment*

Tugas dari *word alignment* adalah menemukan korespondensi antara kata pada teks paralel (Mihalcea dan Pedersen, 2003). Pada dasarnya, *task* ini diperlukan sebagai salah satu proses dalam MT untuk mendapatkan secara otomatis kata-kata yang berpasangan dari korpus paralel dwibahasa atau lebih. Secara umum, proses dari pemasangan kata berlangsung seperti pada gambar 2.3.



Gambar 2.3: Contoh *word alignment*

Kasus yang dapat terjadi pada proses *alignment* ini salah satunya adalah ketika terdapat kata yang tidak memiliki pasangan. Contoh dari kasus tersebut dapat dilihat pada gambar 2.4.



Gambar 2.4: Contoh lain *word alignment*

Bila melihat bahasa Indonesia sebagai sumber bahasa, maka kata "segalanya" pada kalimat tersebut tidak memiliki pasangan dalam bahasa Inggris K1. Pada kasus-kasus seperti ini, biasanya akan ada token khusus yang nantinya akan berpasangan dengan kata-kata yang tidak mempunyai pasangan.

Tool yang digunakan untuk keperluan *word alignment* pada penelitian ini adalah Giza++ (Och dan Ney, 2003). *Tool* tersebut merupakan salah satu *word alignment tools* pada *statistical machine translation* (SMT) yang dapat digunakan untuk memasangkan kata-kata pada dua buah korpus dwibahasa atau lebih. Terdapat beberapa *word alignment tools* lain seperti Berkeley *aligner*, anymalign, dan lain-lain.

2.6 Support Vector Machine

Salah satu *classifier* yang dapat digunakan dalam *supervised learning* untuk melakukan klasifikasi adalah SVM. SVM termasuk sebagai metode klasifikasi yang populer dan telah digunakan untuk berbagai permasalahan seperti klasifikasi teks, *facial expression recognition*, analisis gen, *word sense disambiguation*, dan lain-lain. SVM dapat dikatakan sebagai salah satu metode yang membangun aturan yang dinamakan sebagai *linear classifier* yang secara teori akan menghasilkan kualitas prediksi dari *unseen data* yang baik (Fradkin dan Muchnik, 2006). Pada salah satu penelitian WSD Bahasa Inggris (Zhong dan Ng, 2010), SVM digunakan sebagai *classifier* dari sistem yang dibuat.

Konsep dari cara SVM bekerja adalah dengan menemukan sebuah *hyperplane* dengan *margin* (jarak dari *hyperplane* dengan titik kelas terdekat) yang terbesar. Pemilihan *margin* dengan nilai terbesar ini ditujukan agar *classifier* lebih optimal dalam memisahkan objek dengan kelas yang berbeda. Pada penelitian kali ini, SVM yang digunakan berasal dari *library* Python bernama Scikit (Pedregosa et al., 2011).

2.7 Evaluasi

Precision, Recall, dan F-score merupakan beberapa cara perhitungan untuk merepresentasikan akurasi dari suatu sistem. Berdasarkan domain *information retrieval*, precision merupakan perbandingan dari total jumlah dokumen relevan yang didapatkan oleh sistem dengan total jumlah dokumen yang didapat. Lain halnya dengan precision, recall membandingkan total dokumen relevan yang didapatkan dengan total dokumen relevan dalam *database* (Ting, 2011). Kedua penilaian ini juga digunakan pada sistem yang melakukan klasifikasi suatu *instance*

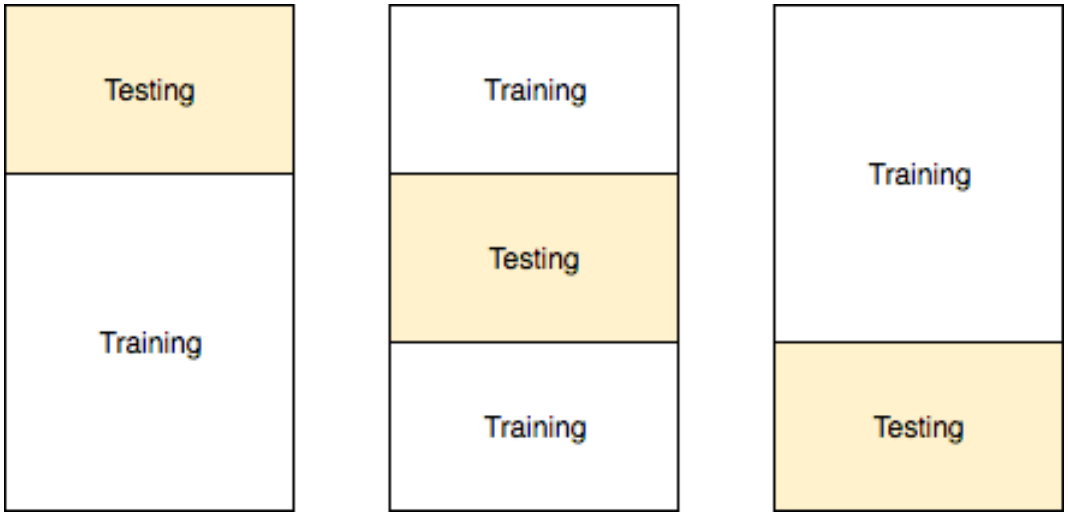
dengan kelas-kelas yang sudah ditentukan.

Pada kasus *word alignment*, terdapat empat buah pengukuran yang dapat dilakukan yaitu *precision*, *recall*, *f-measure*, dan *alignment error rate (AER)* (Mihalcea dan Pedersen, 2003). Diberikan hasil *alignment* dari program berupa A, dan *gold standard alignment* dari *evaluator* (manusia) sebagai G, masing-masing mengandung dua buah *set* yaitu *probable alignment* dan *sure alignment*. Karena *tool alignment* yang digunakan pada penelitian ini menghasilkan *sure alignment* untuk setiap pasangan, maka dilakukan penyesuaian formula penghitungan seperti rumus pada (2.1).

$$\begin{aligned} Precision &= A \cap G/A \\ Recall &= A \cap G/G \\ FScore &= 2PR/P+R \end{aligned} \tag{2.1}$$

Perhitungan pada rumus (2.1) dilakukan setelah anotator melakukan anotasi data yang diberikan dari hasil *random sampling* data *alignment* yang ada. *Random sampling* dilakukan dengan cara megambil sebagian dari keseluruhan data secara acak.

Evaluasi yang dilakukan pada sistem WSD yang dibuat juga menggunakan nilai precision dan recall untuk menghitung F-Score. Pada perhitungan F-Score tersebut, teknik *cross validation* juga digunakan untuk membagi data menjadi *training* dan *testing* data. Training data merupakan bagian dari data keseluruhan yang digunakan untuk memberikan *knowledge* agar nanti mesin dapat melakukan klasifikasi terhadap *input* yang diberikan. Sementara itu, *testing* data merupakan bagian dari data yang akan diprediksi oleh mesin terlebih dahulu (dari hasil latihan dengan *training* data), lalu diperiksa prediksinya dengan kelas sebenarnya pada *testing* data. Metode *cross validation* yang digunakan pada penelitian ini adalah K-Fold *cross validation*. Ilustrasi dari metode tersebut dapat dilihat pada gambar 2.5.



Gambar 2.5: K-Fold *cross validation* dengan nilai k=3

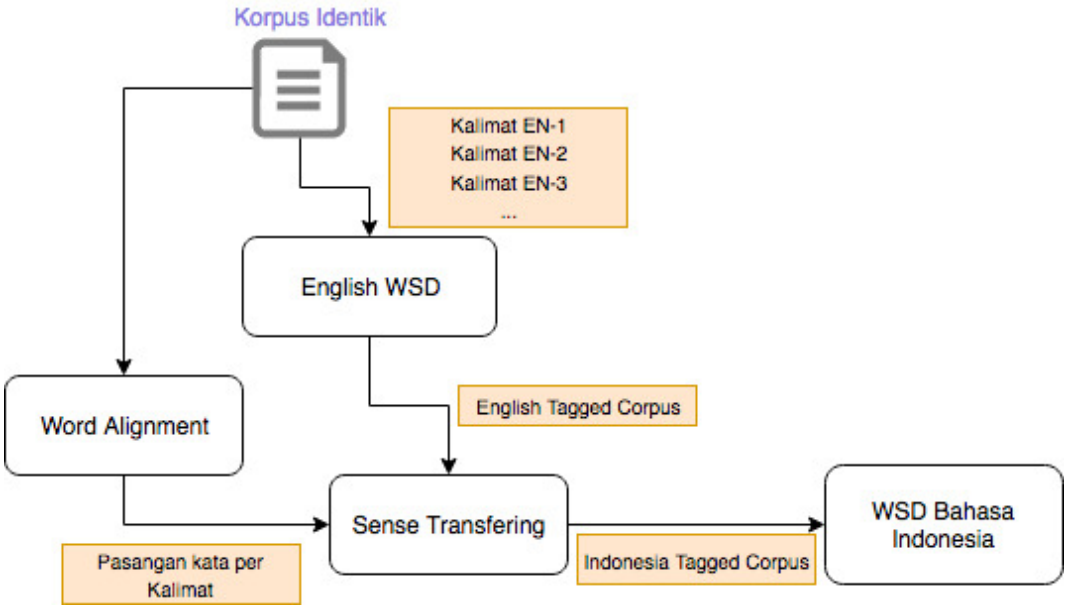
Pada K-Fold *cross validation*, variabel K akan digantikan dengan nilai yang menunjukan pembagian dari *training* dan *testing* yang akan dilakukan. Pada contoh gambar dengan k=3, keseluruhan data akan dibagi menjadi 3 bagian, lalu 1 bagian tersebut akan digunakan untuk *testing* dan sisanya digunakan sebagai *training*. Hal ini dilakukan sebanyak K kali, yang mana pada contoh gambar dilakukan sebanyak 3 kali. Hasil perhitungan F-Score nantinya akan diambil rata-ratanya dari setiap iterasi *cross validation*.

BAB 3
RANCANGAN PENELITIAN

Bab ini akan menjelaskan gambaran proses penelitian secara keseluruhan mulai dari rancangan sistem sampai dengan metode evaluasi. Rancangan dari sistem mempunyai dua hasil utama yaitu *sense tagged corpus* dan sistem WSD Bahasa Indonesia.

3.1 Rancangan Sistem

Rancangan sistem pada penelitian ini dapat dilihat pada gambar 3.1 berikut.

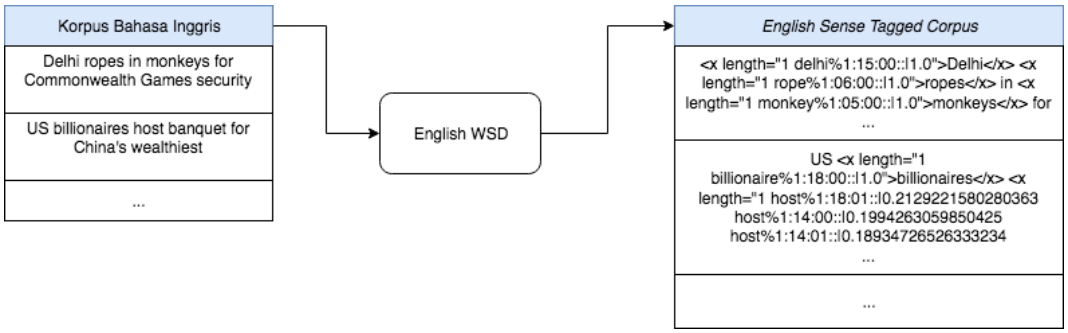


Gambar 3.1: Rancangan Sistem

Proses *pipeline* terdiri dari pembuatan *sense tagged corpus* bahasa Inggris, *word alignment* korpus paralel, peningkatan kualitas dan evaluasi *word alignment*, pemindahan *sense* dari korpus bahasa Inggris, dan sistem WSD yang diimplementasikan. Semua data di dalam korpus identik digunakan dalam proses *word alignment* dan *English* WSD di atas.

3.2 Pembangunan *Sense Tagged Corpus* Bahasa Inggris

Pembangunan *sense tagged corpus* bahasa Inggris dilakukan dengan menggunakan *tool* IMS (Zhong dan Ng, 2010) untuk mendapatkan makna terbaik yang dapat ditag oleh *tool* tersebut. Makna kata hasil dari proses ini akan dipindahkan ke kata yang bersesuaian pada kalimat yang sama pada bagian *sense transfering*. Gambaran proses secara singkat dapat dilihat pada 3.2 berikut.



Gambar 3.2: Ilustrasi Pembuatan *sense tagged corpus* Bahasa Inggris

File yang diberikan sebagai masukan dari IMS adalah korpus Bahasa Inggris yang berisi kalimat-kalimat dalam Bahasa Inggris. Korpus ini didapatkan setelah memisahkan konten dari korpus identik menjadi dua buah korpus Bahasa Indonesia dan Bahasa Inggris. Keluaran dari proses ini adalah korpus berbahasa Inggris dengan kata-kata yang sudah diberikan *tag* berupa *sense key* yang memungkinkan untuk kata tersebut. Hasil tersebut kemudian akan diproses lagi untuk memilih *sense* terbaik dan menyederhanakan format untuk proses berikutnya yang akan dijelaskan lebih rinci pada bab implementasi.

3.3 *Word Alignment* Bahasa Indonesia - Bahasa Inggris

Korpus utama yang digunakan sebagai sumber data penelitian ini adalah korpus identik (Larasati, 2012). Korpus identik berisi pasangan kalimat-kalimat dalam bahasa Indonesia dan Inggris. Kalimat yang berpasangan di dalamnya sebagian besar mempunyai makna konten yang paralel walaupun terdapat juga yang *comparable*. Korpus identik ini mempunyai total sebanyak 88.918 buah pasangan kalimat di dalamnya. Format korpus identik adalah sebagai berikut:

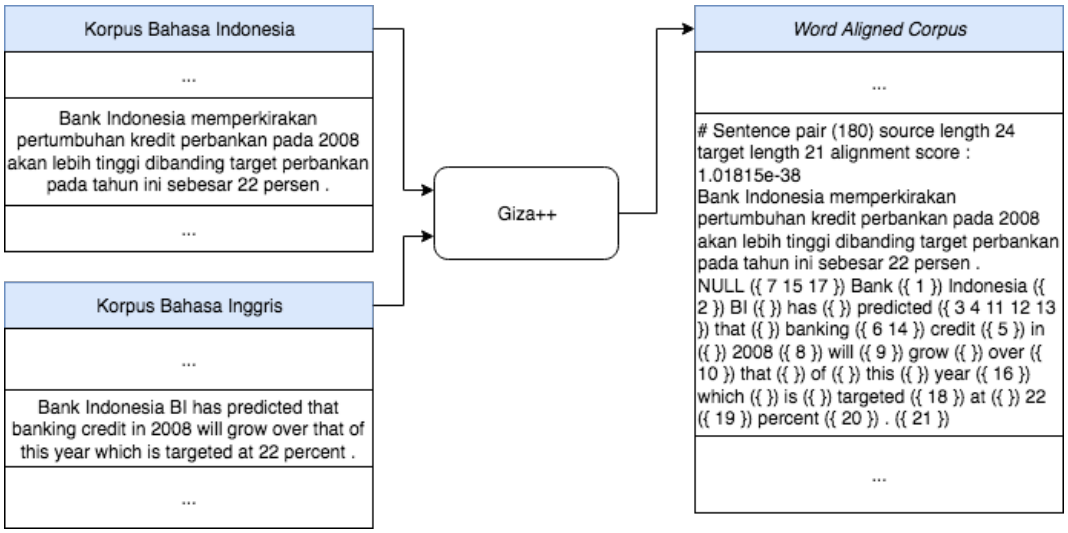
```
<ID><tab><Kalimat Indonesia><tab><Kalimat Inggris>
```

Contoh isi dari korpus identik yang digunakan berbentuk seperti berikut ini:

```
...
```

```
panl-bppt-eco-s1505 Ada 10 sektor yang dibicarakan. There are 10
    sectors that have been on the talk.
panl-bppt-eco-s1508 Semua ketentuan Insya Allah akan selesai 1
    April, kata Burhanuddin. All the regulations, God willing,
    will be completed on April 1, he said.
...
```

Proses *word alignment* ini dilakukan agar nantinya makna kata dari suatu kata dalam Bahasa Inggris dapat ditransfer ke kata yang bersesuaian pada Bahasa Indonesia. Gambaran proses dapat dilihat pada gambar 3.3.



Gambar 3.3: Ilustrasi *alignment* dengan Giza++

Penjelasan dair proses *alignment* tersebut meliputi tahap-tahap berikut:

1. Mempersiapkan kedua buah *file* yaitu korpus bahasa asal (*source*) dan korpus bahasa tujuan (*target*). Kedua *file* ini berpasangan dalam setiap barisnya. Baris pertama dalam *file* pertama berpasangan dengan baris pertama pada *file* kedua sampai akhir baris pada kedua *file*.

2. Menghasilkan *file* perbendaharaan kata dari kedua bahasa dan *list* indeks perbendaharaan kata pada tiap kalimat yang sudah diselaraskan

3. Menghasilkan *cooccurence file* dari kosa kata dan pasangan kalimat tersebut

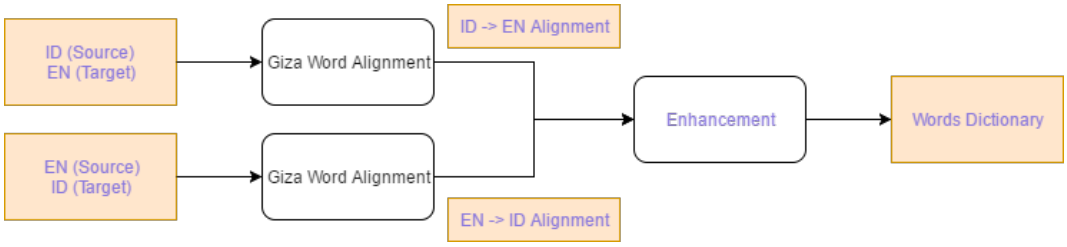
4. Proses *alignment* yang menghasilkan beberapa macam *output file*

Terdapat satu buah *output file* Giza++ yang berisi pasangan-pasangan kalimat dengan kata-kata yang sudah diselaraskan dengan translasinya dalam bahasa tujuan. Hasil ini merupakan *best viterbi alignment* menurut Giza++. Penjelasan lebih

mendalam tentang *output file* Giza++ tersebut akan dibahas pada bab implementasi. Hasil dari proses ini kemudian akan ditingkatkan kualitasnya sebelum melakukan proses *sense transfering*.

Proses peningkatan kualitas hasil alignment diperlukan untuk meminimalisir kesalahan pemasangan kata-kata pada proses sebelumnya. Permasalahan yang terjadi adalah adanya pasangan-pasangan kata yang tidak benar seperti pada halnya kata "lapangan" misalnya yang dipasangkan dengan kata dalam bahasa Inggris yaitu *field*, *ground*, *involved*, *job*, *program*, dan beberapa kata lainnya. Peningkatan kualitas *alignment* ini dilakukan dengan dua buah pendekatan, yaitu dengan bantuan *online dictionary* bahasa Indonesia-Inggris (*crawling*) dan *bi-directional alignment*.

Pendekatan dengan bantuan kamus *online* hasil *crawling* diterapkan dengan mencari kata terjemahan pada bahasa Inggris dari suatu kata dalam Bahasa Indonesia untuk menentukan apakah *alignment* benar atau salah. Pada pendekatan kedua, dilakukan *inverse alignment* antara bahasa Indonesia ke Inggris (*bi-directional*). Jika pada proses awal *alignment* dilakukan dengan menerapkan bahasa Indonesia sebagai *source* dan Inggris sebagai *target*, kali ini dilakukan proses yang berkebalikan. Gambaran proses secara umum pada *bi-directional* ini dapat dilihat pada gambar 3.4.



Gambar 3.4: Bi-directional *Enhancement*

Pemanfaatkan hasil *alignment* korpus bahasa Inggris ke Indonesia akan menghasilkan pasangan-pasangan kata dengan tingkat kesalahan *alignment* lebih kecil dari *alignment* satu arah saja. Metode yang akan dilakukan adalah dengan memeriksa setiap pasangan kata dari bahasa Indonesia yang mana merupakan kata dalam bahasa Inggris, apakah kata tersebut memiliki pasangan dalam *inverse alignment* Giza.

3.4 *Sense Transfering*

Pemindahan makna kata dilakukan dengan beberapa *sub-process* yang terdiri dari pemasangan antar kalimat, pemeriksaan kata, dan *sense transfering*.

1. Pemasangan antar kalimat yang bersesuaian dengan kata-kata yang berpasangan. Pada contoh kata "halaman" yang berpasangan dengan "courtyard", maka pasangan kalimat "Aku bermain di halaman" akan dipasangkan dengan kalimat "I play at the courtyard". Pemeriksaan untuk kata yang saling berpasangan menggunakan hasil dari *alignment* dan kamus hasil *alignment enhancement*.

2. *Sense* dari kata yang menjadi *target* tersebut ("halaman") kemudian diperiksa dengan *sense* kata yang sama yang sudah pernah dipindahkan dari pasangan kalimat lain. Jika *sense* yang ingin dipindahkan "mirip" dan memiliki kedekatan makna diatas batas *threshold*, maka *sense* yang akan dipindahkan hanya salah satunya saja. Proses ini diperlukan untuk meminimalisir adanya satu kata bahasa Indonesia yang mempunyai lebih dari satu *sense* yang mirip dari definisi makna tersebut.

3. Pemindahan makna dilakukan sebagai proses akhir. Bila "courtyard" memiliki *sense* yang artinya adalah "pekarangan rumah", maka "halaman" pada kalimat "Aku bermain di halaman" memiliki *sense* "pekarangan rumah".

3.5 WSD Bahasa Indonesia

Sistem WSD yang dibangun adalah dengan menggunakan pendekatan *supervised learning*. *Classifier* yang digunakan dalam sistem *WSD* ini adalah SVM dengan fitur-fitur tertentu. Pengujian dilakukan dengan menggunakan beberapa fitur seperti *bag of words*, *POS Tag*, dan *word embedding*. Fitur *bag of words* menggunakan *window* sebanyak dua buah kata kanan dan kiri kata tujuan sebagai kata konteks. Fitur *POS Tag* dan vektor *word embedding* juga akan dicoba sebagai skenario pada penelitian ini. Pada ekstraksi fitur, kata-kata yang termasuk dalam stopwords (Tala, 2003) tidak digunakan sebagai fitur.

Fitur *bag of words* menggunakan pendekatan kemunculan kata-kata pada konteks kalimat sebagai fitur. Kata yang diambil untuk dijadikan fitur adalah dua buah kata di kiri dan di kanan dari *target word*. Proses diawali dengan mencari *target word* pada kalimat, kemudian ambil kata sebelum dan sesudah dari *target word* tersebut. Contoh dari fitur ini dapat dilihat pada kalimat dengan *target word* "bisa" berikut.

```
Ani digigit ular dengan bisa yang berbahaya
```

Pada contoh kalimat tersebut, *bag of words* yang diambil adalah "digigit", "ular", dan "berbahaya". Kata "yang" tidak masuk sebagai fitur karena termasuk dalam

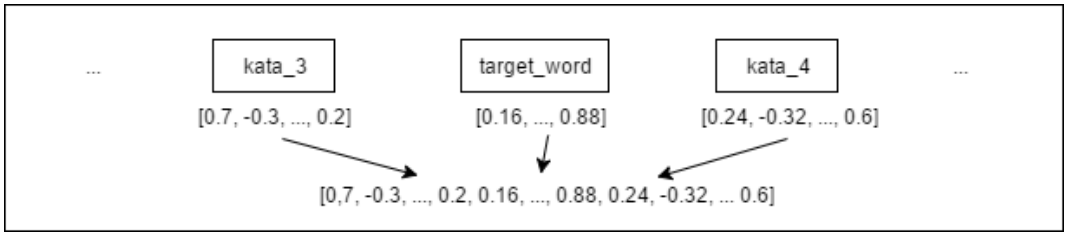
stopwords.

Setiap fitur *bag of words* dari setiap kalimat tersebut dikumpulkan untuk menjadi satu fitur besar yang mendeteksi kemunculan kata-kata tersebut pada setiap kalimat.

Pada fitur POS Tag kelas kata yang diambil adalah POS Tag dari *target word* kata sebelum, dan kata sesudahnya. Kelas kata dari *target word* diperhitungkan karena pada beberapa kasus kata polisemi dapat dibedakan maknanya berdasarkan POS Tag yang dimilikinya. Kelas pada POS Tag yang digunakan meliputi NN(Noun), NNP(Proper Noun), VB(verb), CC(Coordinating conjunction), dan lain-lain. Pembentukan kelas POS Tag untuk menjadi fitur dilakukan dengan proses yang serupa pada fitur *bag of words* dimana kombinasi kombinasi dari POS Tag yang mungkin direpresentasikan dalam *one hot representation*. Hal ini dapat diilustrasikan dengan proses berikut:

1. Simpan POS Tag dari indeks kata masing-masing (*target word*-1, *target word*, *target word*+1)
2. Untuk masing-masing indeks baik itu -1,0,dan 1
3. Kumpulkan POS Tag yang *distinct*, populasikan dalam *array*
4. *Array* ini nantinya akan merepresentasikan keberadaan POS Tag tertentu pada kata dengan indeks terkait tersebut

Pada fitur *word embedding*, vektor yang dijadikan sebagai fitur adalah vektor dari semua kata pada kalimat tersebut. Proses tersebut dapat diilustrasikan pada gambar 3.5.



Gambar 3.5: Ilustrasi Fitur Word Embedding

Semua vektor nilai dari kata-kata tersebut kemudian dikonkatenasi menjadi satu buah vektor besar. Untuk menjamin panjang vektor yang sama untuk setiap kalimat (karena jumlah kata tiap kalimat bisa berbeda-beda), panjang vektor disesuaikan dengan kalimat terpanjang dari semua kalimat yang mengandung *target word* tersebut.

3.6 Evaluasi

Terdapat evaluasi khusus untuk proses *word alignment* dan sistem WSD Bahasa Indonesia yang dibangun. Evaluasi pada bagian *word alignment* ditujukan untuk mengetahui seberapa baik dan *reliable* proses *alignment* yang dilakukan Giza++. Sementara itu, sistem WSD dievaluasi untuk mendapatkan gambaran performa dari fitur seperti apa yang memberikan akurasi terbaik.

3.6.1 *Word Alignment*

Word alignment hasil dari *tool* Giza++ dievaluasi dengan membandingkan hasil *alignment* dari *anotator*(orang yang melakukan evaluasi *alignment* secara manual) dengan hasil *alignment* Giza. Hasil *alignment* yang dilakukan anotator akan dijadikan sebagai *gold standard* acuan untuk perbandingan. Nilai-nilai yang akan dihitung meliputi *precision* (P), *recall* (R), dan F-*score*.

3.6.1.1 Proses Anotasi

Anotator diberikan panduan untuk melakukan anotasi sesuai dengan *task* yang dilakukan Giza. Kedua anotator diminta untuk memasangkan kata-kata yang bersesuaian untuk masing-masing kalimat (dalam bahasa Inggris dan Indonesia) dengan format *file* yang mirip dengan Giza. Panduan anotasi yang diberikan dapat dilihat pada lampiran.

3.6.1.2 Proses Evaluasi

Hasil *alignment* dari kedua anotator masing-masing dibandingkan dengan hasil *alignment* yang dilakukan oleh Giza. Perhitungan Precision, Recall, dan F-score dihitung berdasarkan perbandingan seberapa banyak *alignment* yang benar dengan banyaknya *alignment* yang dilakukan. Proses lebih rinci dari perhitungan yang dilakukan akan dijelaskan pada Bab 4.

3.6.2 Sistem WSD

Hasil dari *sense transfering* kemudian akan digunakan sebagai *training* dan *testing* data untuk menguji performa dari sistem yang dibangun. Sistem ini akan melakukan klasifikasi terhadap kata yang diberikan dengan pengetahuan berupa *sense* apa saja yang mungkin untuk kata tersebut, dan kalimat dimana kata tersebut muncul. Performa dari klasifikasi tersebut akan dilihat berdasarkan perhitungan F1-score

dari hasil klasifikasi makna kata yang dilakukan dengan menggunakan K-Fold *cross validation*. Hasil dari *classifier* tersebut akan dibandingkan dengan baseline berupa *classifier* dengan pendekatan *most frequent class*.

BAB 4
IMPLEMENTASI

Bab ini akan menjelaskan perihal implementasi dari rancangan yang sudah dibuat pada bab sebelumnya, mulai dari pembangunan *sense tagged corpus* bahasa Inggris sampai dengan evaluasi.

4.1 Pembangunan *Sense Tagged Corpus* Bahasa Inggris

Proses pertama yang dilakukan adalah memisahkan kalimat bahasa Inggris dan bahasa Indonesia dari korpus identik menjadi dua buah *file* paralel untuk dapat diproses pada tahap-tahap berikutnya. Korpus yang dihasilkan setelah proses ini adalah dua buah korpus (korpus bahasa Indonesia, dan korpus bahasa Inggris) dimana masing-masing baris merupakan kalimat yang saling berpasangan.

Sense Tagged Corpus dibangun dengan menggunakan bantuan IMS untuk memberikan tag pada korpus bahasa Inggris. *Pre-processing* dilakukan terlebih dahulu terhadap korpus bahasa Inggris seperti menghilangkan tanda baca dan mengubah semua token menjadi huruf kecil. Proses *tagging* dilakukan dengan menjalankan perintah:

Kode 4.1: *Tagging* korpus bahasa Inggris dengan IMS

```
./testPlain.bash <model> <file_input> <file_output> <
   file_index_sense>
```

model yang digunakan IMS pada penelitian ini adalah model yang tersedia pada *website software NUS* berdasarkan versi Wordnet 3.0. Model yang digunakan tersebut meliputi hasil *training* kata-kata dalam bahasa Inggris yang sudah dibangun pada *sense tagged corpus* bahasa Inggris (Taghipour dan Ng, 2015). Proses yang dilakukan IMS dalam melakukan *tagging sense* adalah dengan mengiterasikan melakukan *sentence splitter*, *tokenizing*, *POS Tagging*, dan *lemmatizing*. Setelah proses itu dilakukan, ekstraksi fitur dilakukan sebelum hasilnya masuk ke dalam *classifier* berdasarkan model kata yang sudah ada.

Hasil *output* dari *tool* tersebut adalah korpus dengan kata-kata sudah ditag dengan makna kata yang mungkin(dalam bentuk *sense key*). Makna kata yang diambil untuk kata tersebut merupakan *sense key* dengan nilai kemungkinan terbesar. *Sense key* merupakan *identifier* unik yang menyimpan arti dari suatu kata pada Wordnet Princeton dengan format "lemma_suatu_kata%key". Untuk

mempermudah pemakaian *sense tagged corpus* ini pada proses selanjutnya, dilakukan *post-processing* untuk mengubah hasil keluaran ke dalam format berikut.

```
<sentence>kata-1||sensekey-1 kata-2||sensekey-2 ...</sentence>
<sentence>kata-n||sensekey-n kata-m||sensekey-m ...</sentence>
...
```

Contoh dari kalimat yang sudah diberikan *tag* sampai keluar dari *post-processing* adalah:

```
<sentence>years||year%1:28:01:: animals||animal%1:03:00:: have||have%2:40:04::
caused||cause%2:36:00:: havoc||havoc%1:04:00::</sentence>
```

Pada contoh tersebut, kata years memiliki *sense key* berupa "year%1:28:01::", dimana berdasarkan Wordnet mempunyai arti sebagai '*a period of time containing 365 (or 366) days*'. Tidak semua kata dalam korpus bahasa Inggris ditag oleh IMS, kata-kata sapaan seperti "I", "you", "a" , "the", dan beberapa kata lainnya tidak diberikan *sense key*.

4.2 *Word Alignment*

Proses awal yang diperlukan untuk *alignment* adalah memberikan kedua korpus (Indonesia dan Inggris) ke dalam input Giza++. Setelah menghasilkan *file* hasil *alignment*, dilakukan post-processing untuk menghasilkan kamus yang nantinya dapat digunakan untuk melakukan validasi pasangan kata pada proses *sense transfering*.

4.2.1 Pemrosesan *Word Alignment*

Proses *word alignment* menggunakan dua buah *file* yaitu korpus berbahasa Indonesia dan Inggris yang sudah dipisah dari *pre-processing*. Perintah berikut digunakan untuk melakukan *word alignment* dengan Giza pada penelitian ini:

Kode 4.2: *Word alignment* dengan Giza++

```
./plain2snt.out [source_language] [target_language]

./snt2cooc.out [source_language_vcb_file] [
   target_language_vcb_file] [snt_file] > [coocurrence_file]
```

```
./GIZA++ -S [source_language_vcb] -T [target_language_vcb] -C [
   snt_file] -CoocurrenceFile [cooc_file]
```

Command tersebut dilakukan pada direktori *tool* Giza++. Proses pertama akan menghasilkan tiga buuah *file* yaitu dua buah *file vocabulary* yang berisi nomor indeks dengan kata (bahasa asal, dan bahasa tujuan), dan satu buah *file* snt yang berisi *alignment* dari kalimat. Proses snt2cooc akan menghasilkan *cooccurence file* yang merupakan pasangan-pasangan kalimat untuk dapat diproses Giza.

Giza mengeluarkan beberapa *file* hasil dari proses tersebut. *Output* yang akan digunakan diantaranya adalah *file* bernama A3.final yang merupakan pasangan kalimat dengan kata-kata yang sudah dipasangkan sesuai dengan prediksi terbaik hasil pemrosesan Giza.

4.2.2 Post-Processing

Setelah mendapatkan *file* A3 dari Giza, dilakukan *post-processing* untuk menghasilkan *file* yang dengan mudah dapat diproses untuk melakukan *sense transfering*. Berikut ini merupakan salah satu contoh pasangan kalimat pada *file* A3 keluaran Giza.

```
# Sentence pair (47183) source length 9 target length 9 alignment
    score : 6.85298e-13
Undang-Undang No 14 tahun 2008 tentang Kebebasan Memperoleh
   Informasi
NULL ({ }) Law ({ 1 }) No ({ 2 }) 14 ({ 3 }) of ({ }) 2008 ({ 4 5
    }) on ({ 6 }) Freedom ({ 7 8 }) of ({ }) Information ({ 9 })
```

Pembacaan pasangan kata berdasarkan hasil keluaran dilakukan dengan indeks nomor kata yang berada pada dalam kurung kata di bahasa Inggris. Karena pemisah token *by default* adalah spasi, maka kata "Undang-Undang" adalah kata dengan indeks nomor 1, kata "No" diberikan indeks nomor 2, dan berlaku hal yang sama sampai kata "Informasi". Pada kata bahasa inggris, "Law" dipasangkan dengan indeks satu yang mana adalah "Undang-Undang", kata "No" dipasangkan dengan indeks dua yang mana adalah "No", dan selanjutnya sampai kata *Information*.

Terdapat dua buah *post-processing* yang dilakukan dengan tujuan masing-masing untuk:

1. Penyimpan pasangan kata-kata yang bersesuaian untuk sistem WSD.
2. Sebagai *resource* untuk proses *enhancement word alignment*.

Untuk keperluan nomor satu, bentuk *output* diproses menjadi bentuk lain dengan format:

```
<pair>kata_en_1||kata_id_1 kata_id_2</pair>##<pair>kata_en_2||kata_id_3
</pair>...</pair>
```

Contoh dari hasil *post-processing* pada pasangan kalimat sebelumnya adalah:

```
<pair>law||undang-undang</pair>##<pair>no||no</pair>##<pair
   >14||14</pair>##<pair>2008||tahun 2008</pair>##<pair>on||
   tentang</pair>##<pair>freedom||kebebasan memperoleh</pair>##<
   pair>information||informasi</pair>
```

Hasil ini kemudian disimpan sebagai sebuah *file* sendiri yang akan digunakan kembali pada proses selanjutnya. Sementara itu, keperluan nomor dua difokuskan untuk membuat *dictionary* yang akan ditingkatkan kualitasnya pada tahap berikutnya. Untuk menghasilkan *file* yang dibutuhkan pada nomor kedua, dilakukan pengumpulan pasangan kata bahasa Indonesia dengan bahasa Inggris. Bila misalkan pada kalimat ke-n terdapat kata "undang-undang" yang dipasangkan dengan "law", dan pasangan kata "undang-undang" dengan "regulation" pada kalimat lain (kalimat ke-m, dimana m != n). Berdasarkan kedua kalimat tersebut, maka kata "undang-undang" akan berpasangan dengan dua kata yaitu "law", dan "regulation".

4.3 Peningkatan Kualitas *Alignment*

Hasil dari *alignment* kata yang dilakukan Giza masih menghasilkan pasangan-pasangan kata yang tidak tepat. Untuk mengurangi jumlah pasangan kata yang salah tersebut, dilakukan *enhancement* terhadap hasil pasangan kata dari Giza. Terdapat dua macam peningkatan kualitas *alignment* yang digunakan pada penelitian ini, yaitu *crawling based* dan *bi-directional based*.

4.3.1 Pendekatan *Crawling*

Konsep *crawling* pada penelitian ini mengacu pada kebutuhan untuk *filtering* hasil pasangan kata yang salah berdasarkan kamus Indonesia-Inggris. Karena keterbatasan *resource digital* kamus tersebut, maka dibutuhkan pendekatan *crawling* dari *online dictionary* untuk mendapatkan pasangan kata yang benar. Salah satu *online dictionary* yang dapat diakses dan memiliki hasil terjemahan yang cukup baik adalah *website* sederet.com. *Crawling* dilakukan dengan memeriksa setiap

pasangan bahasa Inggris dari kata Indonesia hasil Giza, apakah pasangan kata tersebut berada pada hasil penerjemahan yang sesuai. Ilustrasi dari proses ini dapat dimodelkan dalam contoh berikut:

1. Kata "halaman" memiliki pasangan bahasa Inggris hasil Giza yaitu "*courtyard*", *"yard"*, *"page"*, dan "*lawn*".
2. Gunakan *crawler* untuk menerima hasil terjemahan dari kata "halaman" dalam bahasa Inggris
3. *Crawler* mendapatkan hasil kata "*page*", dan "*courtyard*".
4. Berdasarkan kedua hasil tersebut, maka pasangan kata "halaman" yang dianggap benar adalah irisan dari kedua *set* tersebut yaitu "*page*" dan "*courtyard*"

4.3.2 Pendekatan *Bi-directional*

Metode lain yang digunakan untuk meningkatkan kualitas *aligment* yaitu dengan memanfaatkan *bi-directional alignment*. Proses yang dilakukan adalah melakukan validasi terhadap kata-kata yang berpasangan dari kedua korpus. *Source code* dari proses *enhancement bi-directional* ini dapat dilihat pada kode berikut.

Kode 4.3: Word Alignment Enhancement

```
dict_id = {}
dict_en = {}


# this section is for filtering which english word that has
    corresponding indo translation (bidirectional) from Giza
    output
for indo_word in dict_id.keys():
  for en_word in dict_id[indo_word].keys():
    if en_word != dict_en:
    # filtering so no same translation is entered, answer ->
        answer, jawaban -> jawaban
      if en_word in dict_en and indo_word in dict_en[en_word] and
          en_word not in stop:
        if indo_word not in final_dictionary:
          final_dictionary[indo_word] = { en_word: dict_en[
              word_en][word_id] }
        else:
```

```
            if en_word not in final_dictionary[indo_word]:
              final_dictionary[indo_word][en_word] = dict_en[
                 word_en][word_id]
```

Pertama, setiap kata dalam bahasa Indonesia dikumpulkan terlebih dahulu dengan setiap pasangan kata bahasa Inggrisnya (satu kata bisa memiliki lebih dari satu pasangan) ke dalam variabel dict_id. Proses selanjutnya adalah melakukan pengumpulan yang serupa terhadap kata dalam bahasa Inggris dengan pasangan kata dalam bahasa Indonesianya (pada variabel dict_en). Berbagai kata dalam bahasa Indonesia beserta pasangannya disimpan sebagai "kamus-1 (dict_id)". Kata dalam bahasa Inggris, beserta pasangan kata Indonesianya disimpan sebagai "kamus-2"(dict_en). Variabel stop berisi *list* of stopwords Bahasa Inggris dari *library* NLTK. Proses validasi kebenaran dari pasangan kata dilakukan dengan cara:

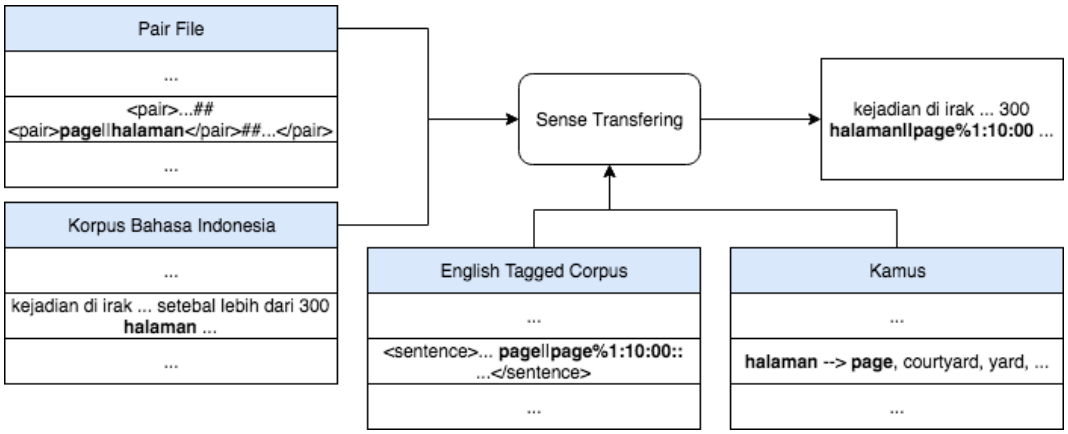
1. Untuk setiap kata di bahasa Indonesia dalam "kamus-1" semisal kata "kali".
2. Lakukan pengecekan terhadap setiap pasangan kata di bahasa Inggris pada "kamus-1" dari kata "kali", misalkan pasangan kata bahasa inggrisnya adalah "time", "river", dan "fire".
3. Jika pada "kamus-2" kata "time" dipasangkan dengan "kali", dan "waktu" maka kata "time" merupakan pasangan yang dianggap benar (karena kata "time" berpasangan dengan "kali"). Pada kasus kata "fire", bila pasangan bahasa Indonesianya adalah "api" dan "tungku", maka kata "fire" dianggap bukan pasangan yang tepat dengan "kali" karena tidak terdapat pasangan "fire -> kali".

4.4 *Sense Transfering*

Hasil dari proses peningkatan kualitas *alignment* adalah sebuah "kamus" bahasa yang akan digunakan sebagai referensi untuk memindahkan makna kata dari bahasa Inggris ke kata yang benar pada bahasa Indonesia. Proses ini dilakukan dengan tahap-tahap sebagai berikut:

1. Iterasi untuk setiap kata dalam bahasa Indonesia pada kamus
2. Iterasi pada setiap pasangan kalimat
3. Periksa apakah "*pair*" kata bahasa Indonesia tersebut terdapat di dalam kamus
4. Jika "pair" tersebut benar berada dalam kamus, maka pindahkan makna kata dari *english word* yang bersesuaian.

Ilustrasi dari proses tersebut pada kata "halaman" secara garis besar dapat dilihat pada gambar



Gambar 4.1: Ilustrasi *sense transfering*

Proses yang terjadi pada ilustrasi di atas adalah:

1. Jika misalkan kata "halaman" pada kamus memiliki pasangan kata "page" dan "courtyard".

2. Pada sebuah kalimat "... 300 halaman ..." dimana "*pair*" pada kalimat tersebut diantaranya adalah

```
..<pair>second||kedua</pair>##<pair>page||halaman</pair>..
```

3. Pasangan kata yang didapat dari "halaman" dari *pair* tersebut adalah "page". Berdasarkan hasil tersebut kata "page" kemudian diperiksa pada kamus yang ada.

4. Karena kata "page" merupakan salah satu terjemahan untuk kata "halaman", maka pindahkan makna "page" dari *sense tagged corpus* kalimat tersebut ke kata "halaman" pada kalimat Indonesia dengan indeks yang sama. Hal tersebut dapat dicontohkan pada pemindahan makna "page%1:10:00::" dari contoh *sense tagged corpus* berikut.

```
<sentence>... page||page%1:10:00:: ...</sentence>
```

   Sense key "page%1:10:00::" tersebut kemudian dipindahkan ke kata "halaman" pada kata "halaman" di kalimat bahasa Indonesia yang bersesuaian.

5. Dalam proses pemindahan makna tersebut, akan dilakukan pemeriksaan apakah untuk kata yang sama, terdapat makna kata yang serupa dari hasil transfer kalimat lain.

6. Jika "kemiripan" makna kata yang akan dipindahkan melebihi threshold (0.5), makan akan digunakan *sense key* yang lama karena kedua *sense key* dianggap mempunyai makna yang sama. Penghitungan seberapa dekat "makna" dari kedua *sense key* tersebut dilakukan dengan bantuan *method* path_similarity dari *tool* NLTK dengan korpus wordnet.

Setelah didapatkan setiap kata beserta makna hasil *transfernya*, semua kata tersebut disusun ke dalam bentuk JSON dengan bentuk *array of* kalimat yang isinya adalah *array of words*. Setiap *instance* dari *word* tersebut menyimpan kata, POS Tag, dan *sense key* hasil *sense transfering* jika ada.

4.5 WSD Bahasa Indonesia

Sistem WSD dibangun dengan menggunakan *supervised learning*. Pada sistem ini terdapat tiga buah bagian utama yaitu pemilihan fitur serta *classifier*, ekstraksi fitur, dan evaluasi hasil *classifier*. Proses yang pertama kali dilakukan oleh sistem adalah membaca korpus dan memilih *target word*, mendapatkan *sense key* dari kata tersebut untuk dijadikan sebagai *class* dari klasifikasi. Pada sistem WSD yang dibuat, kata-kata yang terdapat pada *list of stopwords* dihilangkan dari kalimat-kalimat pada data.

4.5.1 Pemilihan Fitur dan *Classifier*

Terdapat tiga buah fitur pada penelitian ini yang terdiri dari:

1. Fitur *bag of words*

2. Fitur *POS tagging*

3. Vektor dari hasil *word embedding*

Classifier yang digunakan pada penelitian ini adalah SVM dari Scikit (Pedregosa et al., 2011) dengan parameter *default* berupa kernel linear dan C=1. Penggunaan SVM pada penelitian ini mengikuti *classifier* yang digunakan pada IMS Zhong dan Ng (2010) untuk melihat performa yang dihasilkan pada sistem WSD Bahasa Indonesia yang dibuat.

4.5.2 Ekstraksi Fitur

Proses yang dilakukan dalam pengambilan kata konteks untuk fitur *bag of words* dilakukan dengan tahap-tahap berikut.

Kode 4.4: Fitur Bag of Words

```
bag_of_words []

for each sentence
split sentence by whitespace into words
for each word
if word == target word
for x in [-2,-1,1,2]
if word(x) exist and word(x) not in bag_of_words
add word(x) into bag_of_words

return bag_of_words
```

Pada *source code* tersebut, fitur *bag of words* diinisiasi dengan *list* kosong. Proses selanjutnya adalah dengan mengiterasi setiap kata dalam setiap kalimat, jika kata tersebut merupakan *target word* yang dicari, ambil kata dengan indeks -2, -1, 1, dan 2 dari kata *target word* sekarang (jika ada). Hasil dari proses tersebut adalah *list* yang berisi kata-kata yang pernah muncul sebagai tetangga dari *target word*. Kumpulan kata-kata tersebut kemudian digunakan untuk mengubah suatu kalimat yang mengandung *target word* menjadi vektor yang merepresentasikan keberadaan kata-kata pada *list bag of words* tersebut.

Fitur POS Tagging menggunakan *tool* dari Stanford bernama "A Part-Of-Speech Tagger" yang dilatih dengan menggunakan model untuk bahasa Indonesia (Dinakaramani et al., 2014). Proses *tagging* ini dilakukan sebelum memasuki sistem WSD terhadap keseluruhan kalimat dalam korpus identik yang berisi bahasa Indonesia saja. Terdapat *pre-processing* awal pada korpus bahasa Indonesia tersebut untuk menghilangkan beberapa tanda baca seperti titik, koma, tanda tanya, tanda seru, dan beberapa tanda baca lainnya. Setelah proses tersebut, diberikan tanda baca berupa titik pada akhir kalimat sebagai indikator akhir sebagai kebutuhan dari kompabilitas *tool* (tidak semua kalimat pada korpus memiliki tanda baca akhir kalimat). Perintah yang dilakukan untuk melakukan *POS Tagging* tersebut adalah:

Kode 4.5: Stanford POS Tagger

```
java -mx300m -cp 'stanford-postagger.jar:lib/*' edu.stanford.nlp.
   tagger.maxent.MaxentTagger -model <model_bahasa_indonesia> -
```

```
    textFile <korpus_bahasa_indonesia>
```

Hasil dari proses tersebut merupakan korpus dengan setiap kata yang sudah memiliki POS Tag dengan format:

```
<kata_1>_<postag_1> <kata_2>_<postag_2> ... <kata_n>_<postag_n>
<kata_x> <postag_x> <kata_y>_<postag_y> ... <kata_z>_<postag_z>
...
```

Proses dalam melakukan eksetraksi fitur POS Tag tersebut dapat dilihat pada 4.6 berikut.

Kode 4.6: Ekstraksi Fitur POS Tag

```
pos_tag_features = [[],[],[]]

foreach sentence in pos_tagged_sentences:
  foreach word_with_postag in sentence:
    word, postag = split_postag(word_with_postag)
    if word == target_word:
      if word_before(word,-1) exist:
        pos_tag_features[0].append(postag(word,-1))
      pos_tag_features[1].append(postag)
      if word_after(word,1) exist:
        pos_tag_features[2].append(postag(word,1))
```

Pada proses ekstraksi fitur POS Tag, dilakukan iterasi untuk setiap kata pada setiap kalimat. Jika *target word* telah ditemukan, ambil POS Tag dari kata sebelum *target word*, *target* word, dan sesudah *target word*. pos_tag_features[0] merepresentasikan POS Tag yang muncul pada sebelum *target word*, pos_tag_features[1] untuk POS Tag *target word*, dan pos_tag_features[2] untuk POS Tag kata sesudah *target word*. Setelah POS Tag masing-masing posisi dikumpulkan, vektor representasi fitur akhir akan bernilai satu jika POS Tag muncul pada kalimat dan nol jika POS Tag tidak muncul.

Fitur ketiga adalah penggunaan vektor dari model *word embedding*. Ekstraksi fitur ini dilakukan dengan proses 4.7.

Kode 4.7: Ekstraksi fitur *word embedding*

```
model = Word2Vec.load(word_embedding_model)

length_we_feature = len(model['yang'])

max_length = max(length(sentences))
```

```
foreach sentence in sentences_containing_target_words:
  arr = []
  words = get_words(sentence)
    for x in range(max_length):
      if x in range of len(words):
        if words[x] in model:
          arr = arr + model[words[x]]
        else:
          arr = arr + array_of_zero(length_we_feature)
      else:
        arr = arr + array_of_zero(length_we_feature)
```

Pada proses tersebut, *length_we_feature* diisi dengan jumlah kata terbanyak pada kumpulan kalimat. Untuk setiap kalimat, setiap kata di dalamnya akan diperiksa terlebih dahulu apakah terdapat vektor kata tersebut pada model. Jika vektor kata tersebut ada pada model *word embedding* yang digunakan, konkatenasi vektor dengan arr yang merepresentasikan fitur pada kalimat tersebut. Namun demikian, jika kata tersebut tidak terdapat pada model ataupun indeks kata sudah melebihi jumlah kata pada kalimat, konkatenasi arr dengan *array* yang berisi angka nol sebanyak panjang dari vektor *word embedding*.

Fitur terakhir yang dicoba adalah gabungan dari fitur *bag of words* dengan *POS Tag*. Implementasi yang digunakan sama dengan ekstraksi fitur *bag of words* dan POS Tag, yang kemudian kedua buah vektor dikonkatenasi menjadi satu buah vektor gabungan.

4.6 Evaluasi dan Anotasi

Proses evaluasi meliputi *pre-processing*, anotasi data, dan evaluasi. Hasil dari evaluasi ini ditujukan untuk memberikan informasi seberapa baik performa dari *alignment* yang dilakukan oleh *tool* Giza. Terdapat beberapa proses dalam mempersiapkan data untuk evaluasi oleh anotator.

1. Pertama, pada setiap kata dalam bahasa Indonesia diberikan sebuah tanda indeks berupa angka untuk mempermudah proses evaluasi anotator nantinya. Pada proses tersebut, kalimat "Aku ingin makan" sebagai perumpamaan, diubah menjadi "Aku(1) ingin(2) makan(3)". Angka tersebut diperuntukan untuk mempercepat dan mempermudah kerja anotator nanti untuk melihat pasangan kata dari kalimat bahasa Inggris.

2. Kedua, kosongkan nomor indeks hasil *alignment* Giza pada kalimat bahasa

Inggris. Perumpamaan pada kalimat "NULL ({ }) i ({ 1 }) want ({ 2 }) to ({ }) eat ({ 3 })" akan berubah menjadi "NULL ({ }) i ({ }) want ({ }) to ({ }) eat ({ })" yang nantinya akan diisi secara manual oleh anotator.

4.6.1 Anotasi

Setelah diberikan pasangan kalimat untuk dianotasi, anotator melakukan anotasi sesuai dengan panduan yang diberikan. Hasil dari anotasi tersebut kemudian dikumpulkan kembali untuk dievaluasi.

4.6.2 Evaluasi *Word Alignment*

Evaluasi perhitungan *precision*, *recall*, f-score, dan *agreement* merupakan penyesuaian dari rumus pada (Mihalcea dan Pedersen, 2003). Perhitungan dilakukan secara otomatis dengan program yang dibuat dengan algoritma berikut.

Kode 4.8: Word Alignment Evaluation

```
def evaluate_bracket(list_giza, list_anotator):
# evaluate per character
numbers_anotator = len(list_anotator)
numbers_giza = len(list_giza)
match = 0
if numbers_giza < numbers_anotator:
# iterate through the numbers giza
for n in list_giza:
if n in list_anotator:
match += 1
else:
# iterate through the numbers anotator
for n in list_anotator:
if n in list_giza:
match += 1
return (numbers_anotator, numbers_giza, match)

# given sentence which already been alignned from the first
    anotator (an1), second anotator(an2), and giza(giza)
matches, total_giza, total_anotator, total_giza_1,
    total_anotator_1 = 0, 0, 0, 0, 0
precision, recall = [[],[]], [[],[]]

for each sentence in sentences:
for each token in sentence:
```

```
(numbers_anotator, numbers_giza, match) = evaluate_bracket(an1(
    token), giza(token))
(numbers_anotator_1, numbers_giza_1, match_1) = evaluate_brakcet(
    an2(token), giza(token))
agreement = count_agreement(an1(token), an2(token))
matches += match
matches_1 += match_1
total_giza += numbers_giza
total_giza_1 += numbers_giza_1
total_anotator += numbers_anotator
total_anotator_1 += numbers_anotator_1
agreement.append(agreement)
precision[0].append(matches/total_giza)
recall[0].append(matches/total_anotator)
precision[1].append(matches_1/total_giza_1)
recall[1].append(matches_1/total_anotator_1)

precision = average(precision[0])
recall = average(recall[0])
precision_1 = average(precision[1])
recall_1 = average(recall[1])
f1 = 2 * precision * recall / (precision + recall)
f2 = 2 * precision_1 * recall_1 / (precision_1 + recall_1)
agreement = average(agreement)
```

Berdasarkan cara perhitungan evaluasi tersebut, terdapat dua buah *precision* dan *recall* yang mana masing-masing menunjukan indikator penilaian untuk anotator pertama dan kedua. Precision didapatkan dengan menghitung jumlah *alignment* yang sama antara anotator dengan hasil giza, dibagi dengan jumlah *alignment* pada Giza. Sementara itu, Recall dihitung dengan cara jumlah *alignment* sama dibagi dengan jumlah *alignment* pada Anotator. Nilai F-score dihitung dengan cara mengalikan 2 dengan Precision dan Recall, lalu membaginya dengan jumlah Precision dan Recall. Precision, Recall, dan F-score masing-masing kalimat kemudian dirata-rata untuk mendapatkan nilai akhir. Agreement pada kedua anotator dilihat dengan membandingkan antara jumlah kesamaan *alignment* pada kedua anotator dibagi dengan jumlah *alignment* anotator pertama.

4.6.3 Evaluasi Sistem WSD Bahasa Indonesia

Target word untuk evaluasi dipilih secara manual yang memenuhi kriteria bahwa kata tersebut memiliki kata *translation* lebih dari satu dengan makna masing-masing yang berbeda. Evaluasi dilakukan dengan 4-Fold *cross validation* menggunakan

perhitungan F1-score dari hasil klasifikasi yang dilakukan SVM terhadap *target word*. Perintah pada *script* yang digunakan untuk inisialisasi *classifier* dapat dilihat pada potongan kode 4.9 berikut.

Kode 4.9: Inisialisasi SVM, baseline

```
clf = svm.SVC(kernel='linear')
bf = dummy.DummyClassifier(strategy='most_frequent')
```

Cross validation (CV) dilakukan dengan 4-Fold CV dengan mengambil hasil akurasi (F-Score) rerata dari tiap *fold*. Sebuah algoritma sederhana digunakan sebagai *baseline* untuk pembanding dari sistem WSD yang dibangun. Baseline menggunakan pendekatan *most frequent* sebagai cara untuk menentukan makna terbaik dari suatu kata. Baseline tersebut akan memberikan kelas makna kata pada semua *target word* sesuai dengan makna kata terbanyak pada *training data*.

BAB 5
HASIL DAN ANALISIS

Bab ini menjelaskan mengenai hasil yang didapatkan dari eksperimen, serta evaluasi dan analisis terkait hasil tersebut.

5.1 *Sense Tagged Corpus* Bahasa Inggris

Tabel 5.1 menunjukan jumlah token (kata) pada korpus bahasa Inggris dan yang diberikan *tag sense* oleh IMS

Tabel 5.1: Jumlah *instance* korpus bahasa Inggris

No	Tipe	Jumlah
1	Token (kata)	1.801.484
2	Kata yang diberikan *tag* oleh IMS	1.024.797

Berdasarkan proses pembuatan dan hasil dari *sense tagged corpus* tersebut, dapat dilihat bahwa tidak semua kata diberikan *sense* oleh IMS. Kata-kata sapaan seperti "I", "you", dan kata *articles* yaitu "a", "the", "an". Tingkat kebenaran dari *sense tag* yang diberikan bergantung dari model yang digunakan pada penelitian. Terdapat banyak kasus-kasus dimana pemberian *sense* yang dilakukan adalah benar seperti misalnya pada kata "*visitor*" yang diberikan *tag* dengan *sense key* 1:18:00::, yang mana berdasarkan wordnet Princeton "visitor%1:18:00::" memiliki arti sebagai "*someone who visits*". Contoh lain dari kata yang diberikan *tag* dengan benar adalah "*company*" pada konteks potongan kalimat "*Plantation company PT* ...". Kata "*company*" tersebut diberikan tag "company%1:14:01::" yang berdasarkan wordnet Princeton memiliki makna "*an institution created to conduct business*". Namun demikian, terjadinya kesalahan pemberian *tag* pada kata terjadi pada kasus-kasus seperti:

1. Sebuah entitas diberikan *tag* dimana entitas tersebut dianggap kata biasa. Contohnya adalah kata "Scotland Yard" dimana "Yard" pada kata tersebut diberikan *tag* yang diartikan sebagai "*a unit of length equal to 3 feet*". Hal ini menunjukan bahwa *tool* belum dapat membedakan antara entitas yang

memang tidak perlu diberikan *tag* dan kata biasa (walaupun kata tersebut sudah memiliki huruf kapital).

2. Kesalahan *tag* dikarenakan *training data* yang digunakan oleh model. Pada potongan kalimat "... *FASB rule will cover such financial instruments as interest rate swaps financial ...*, kata "*interest*" diberikan tag dengan makan "a sense of concern with and curiosity about someone or something". Berdasarkan konteks kalimat tersebut, dapat diketahui bahwa makna yang seharusnya didapat untuk kata "*interest*" di atas ialah "bunga bank". Percobaan untuk *tagging* sense dari kalimat "bank interest is high" juga menghasilkan *sense key* yang sama untuk kata *interest* (mendapatkan *sense key "keingintahuan"*) walaupun pada kalimat tersebut makna yang seharusnya didapat adalah "bunga bank".

3. Pemberian *tag* pada *multi word* token seperti "*make up*" masih diberikan pada setiap kata ("*make*" dan "*up*" memiliki *sense key* masing-masing). Hal ini terjadi karena IMS mengolah kata demi kata dengan proses tokenisasi *by default* menggunakan spasi. Setelah dilakukan pemeriksaan pada kata-kata yang terdapat pada model, kata *make up* ternyata disimpan sebagai "make_up". Berdasarkan pemeriksaan tersebut, dapat disimpulkan bahwa IMS dapat melakukan *tagging* yang benar dari kata "make up" jika dijadikan satu buah token berupa "make_up". hasil *tagging* yang benar untuk kasus seperti kata-kata tersebut memerlukan *pre-processing* untuk mengganti *separator* kata multiword yang umumnya menggunakan spasi dengan "_" agar IMS dapat memberikan *tag multi word* tersebut dengan benar. Selain *pre-processing*, IMS juga dapat melakukan *tagging* dengan input dalam format XML. Bentuk kata-kata dan kalimat dalam format XML tersebut biasanya memiliki *multi word* yang sudah dijadikan satu token sehingga mempermudah penyelesaian masalah tersebut.

5.2 *Word Alignment*

Proses alignment dilakukan terhadap semua pasangan kalimat pada korpus identik (total sebanyak 88.919 buah). Hasil dari proses *word alignment* yang dilakukan Giza dibandingkan dengan hasil *alignment* yang dibuat oleh dua orang anotator. Jumlah yang akan dibandingkan adalah 200 buah pasangan data yang didapat dengan *random sampling*. Indikator performa dari perbandingan tersebut adalah nilai dari *precision*, *recall*, dan F-score sesuai dengan penyesuaian cara perhitungan

dari penelitian (Mihalcea dan Pedersen, 2003). Hasil evaluasi yang didapatkan dapat dilihat pada tabel 5.2

Tabel 5.2: Evaluasi Word Alignment

Anotator	**Precision**	**Recall**	**F-Score**
1	0.775	0.747	0.761
2	0.768	0.75	0.759

Agreement antara kedua anotator pada *random sampling alignment* adalah 0.924.

Evaluasi hasil anotasi tersebut juga dilihat dari sisi perbandingan jumlah *alignment* yang *exact match* dengan total kata. Suatu *alignment* dikatakan sebagai *exact match* jika pasangan kata yang diberikan oleh kedua anotator sama persis. Contoh yang tidak *exact match* adalah jika misalkan kata "keluar" dipasangkan dengan "*get out*" pada anotator 1, namun dipasangkan dengan "*out*" pada anotator 2. *Alignment* dikatakan *exact match* pada kasus sebelumnya jika anotator 2 juga memasangkan kata "keluar" dengan "*get out*". Jumlah dan perbandingan *exact match* antara kedua anotator dengan total pasangan kata dapat dilihat pada tabel 5.3.

Tabel 5.3: Perbandingan *exact match alignment* anotator

***Exact match* (e)**	**Total *alignment* (t)**	**Rasio (e/t)**
4.315	5198	0.83

Perbandingan antara jumlah *alignment* yang *exact match* antara kedua anotator dengan pasangan kata pada 200 *sample* yang diberikan adalah 0.83. Hal ini berarti 83% dari *alignment* yang dilakukan kedua anotator memiliki kesamaan pasangan kata yang identik.

Berdasarkan hasil beberapa analisis *word alignment* tersebut, dapat dilihat bahwa Giza++ secara umum dapat menghasilkan *alignment* pada korpus identik dengan nilai F-Score sekitar 75%. Namun demikian, kesalahan yang terjadi pada *alignment* Giza ada di kasus-kasus seperti:

1. Pemasangan frasa kompleks dengan kata lain. Seperti misalnya frasa "*a very awesome and huge building*" dengan "gedung raksasa", ataupun gaya bahasa lainnya yang menjadikan frasa tersebut kompleks dan panjang.

2. Pasangan kalimat yang tidak sepenuhnya paralel (*comparable*), seperti misalnya "I'm glad that you're here" dengan "Aku senang kau ada". Variasi dari bentuk kalimat *comparable* terkadang mempersulit model dari Giza++ untuk menentukan pasangan kata yang benar.

Walaupun akurasi yang didapatkan sekitar 70%, tidak jarang ditemukan pasangan kata-kata yang tidak sesuai dan dapat menjadi masalah pada saat *sense transfering* sehingga dilakukan proses *word alignment enhancement*.

Hasil *enhancement* dengan pendekatan *bi-directional* masih menghasilkan kata-kata yang memiliki pasangan tidak sesuai seperti misalnya kata "pembuat" dengan "*jewelry*", kata "pendidikan" dengan "*tended*", kata "harapkan" dengan "*anticipate*", dan lain-lain. Skenario pada *enhancement* menggunakan *crawling* kamus Indonesia-Inggris tidak menghasilkan pasangan yang salah seperti pada contoh yang diberikan di atas. Evaluasi terhadap hasil *enhancement* ini dilakukan dengan mengambil 10 buah *sample* secara *random* dari pasangan kata yang ada. Hasil *sampling* tersebut dapat dilihat pada 5.4.

Tabel 5.4: Pasangan kata hasil *random sampling* dengan pendekatan *bi-directional* dan *crawling*

Kata	***Bi-directional***	***Crawling***
memodernkan	*modernize*	*modernize*
kembar	*twin*, *twins*	*twin*, *twins*
ingat	*remember*, *remembered*, *recall*, *note*, *recalls*, *strike*	*remembered*, *recall*, *remember*
peroksida	peroxide	peroxide
imbang	*draw*, *drawing*	*draw*, *drawing*
tegang	*tense*, *edge*	*tense*
berobat	*medical*, *purpose*, *treatment*	*treatment*
membantu	*bolster*, *supporting*, *beneficial*, *helping*, *help*, *support*, *assisted*, *assist*, *appearance*, *assisting*, *helped*, *assists*, *helps*, *aiding*, *aid*, *facilitates*, *supports*, *assistance*	*helped*, *assisting*, *facilitating*, *help*, *assisted*, *assist*, *helpful*, *helping*, *helps*, *aid*, *facilitates*, *supports*, *avail*
disusun	*composed*, *arranged*, *developed*, *structured*, *organized*, *prepared*, *compiled*	*composed*, *arranged*, *developed*, *structured*, *prepared*, *compiled*
diusulkan	*suggested*, *suggests*, *recommended*, *proposed*, *proposes*	*suggested*, *proposed*

Pasangan kata pada tabel di atas kemudian dihitung berapa jumlah pasangan yang benar untuk masing-masing pendekatan (*bi-directional* dan *crawling*). Pasangan kata dianggap benar jika hasil translasi kata tersebut dapat digunakan untuk konteks kalimat tertentu. Hasil evaluasi dengan mengukur precision dari hasil tersebut dapat dilihat pada tabel 5.5.

Tabel 5.5: Precision dari hasil *random sampling alignment enhancement*

Kata	***Bi-directional***	***Crawling***
memodernkan	1.0	1.0
kembar	1.0	1.0
ingat	0.67	1.0
peroksida	1.0	1.0
imbang	0.5	0.5
tegang	1.0	1.0
berobat	0.33	1.0
membantu	0.73	0.84
disusun	0.71	0.66
diusulkan	0.6	1.0
Rata-rata	0.78	0.90

Berdasarkan evaluasi pada tabel di atas, didapatkan bahwa pendekatan *crawling* pada *random sampling* tersebut memiliki tingkat precision yang lebih tinggi dari *bi-directional* yaitu sebesar 90%. Kamus hasil dari *crawling* inilah yang akan digunakan untuk proses *sense transfering*.

5.3 *Sense Transfering*

Proses *transfer* makna kata dari bahasa Inggris ke bahasa Indonesia yang dilakukan sangat bergantung dari hasil *alignment* kata pada proses sebelumnya. Untuk sebagian besar kata yang memiliki pasangan kata yang benar, proses *transfer* dapat menghasilkan makna yang benar juga. Hal tersebut didukung jika *sense tagged word* pada korpus bahasa Inggris juga benar). Terdapat beberapa kata yang dipilih sebagai *sampling* untuk mengevaluasi hasil *sense transfering*. Kelompok ini dibagi menjadi:

1. Jumlah Kelas
   (a) 3-5 kelas kata
   (b) lebih dari 5 kelas kata
2. sebaran jumlah *instance* dalam kelasnya
   (a) *balance*
   (b) *imbalance*

3. Bentuk morfologi dari kata tersebut

   (a) Lemma (kata dasar)

   (b) Berimbuhan baik itu infleksional ataupun *derivative*

Pada laporan ini, hasil dari *sense transfering* yang dianalisis adalah proses yang menggunakan kamus *crawling*. Pada jumlah kelas sebanyak 3-5 kelas kata (*sense key*), *target word* yang diambil adalah "memecahkan". Kata tersebut memiliki 4 buah kelas total dengan *sense key* yang didapat yaitu ’solve%2:31:00::’,’resolve%2:31:01::’, ’break%2:30:03::’, dan ’split%2:38:00::’. Kata "menolak" mewakili kelas kata sebanyak 9 buah yang diantaranya mengandung kelas ’reject%2:40:00::’, ’decline%2:32:00::’, rebuff%2:32:00::, dan enam buah kelas lainnya. Makna kata ’decline%2:32:00::’ sendiri sebenarnya merupakan "gabungan" dengan ’refuse%2:32:00’ yang memiliki kemiripan dengan nilai *path similarity* melewati batas *threshold*, sehingga makna kata dianggap sama dan diberikan *tag* sebagai ’decline’ (dipilih satu *tag* saja). Tabel 5.6 menunjukan contoh beberapa kata tersebut dalam beberapa konteks kalimat yang bersesuaian.

Tabel 5.6: Evaluasi *Sense Transfering* Berdasarkan Jumlah Kelas

Sense Key	**Makna**	**Kalimat**
solve%2:31:00::	*find the solution to (a problem or question) or understand the meaning of*	salah satu cara untuk **memecahkan** persoalan yang pelik...
resolve%2:31:01::	*bring to an end / settle conclusively*	evolusionis masih belum bisa **memecahkan** perma-salahan darwin...
break%2:30:03::	*terminate*	...base mereka **memecahkan** rekor untuk...
split%2:38:00::	*go one's own way; move apart;*	senat mereka **memecahkan** perbedaan antara skenario 1 dan 3 dengan
decline%2:32:00::	*show unwillingness towards*	wells rich **menolak** untuk berkomentar...
rebuff%2:32:00::	*reject outright and bluntly*	georgia gulf **menolak** penawaran tersebut ...
reject%2:40:00::	*refuse to accept*	..dua pekan lalu **menolak** tawaran pemerintah...

Makna kata *split* yang diberikan hanya berjumlah satu buah dari keseluruhan korpus, hal ini disebabkan karena kata bahasa Inggris yang digunakan pada kalimat bahasa Inggrisnya menggunakan kata *split*. Berdasarkan *sampling* yang dilakukan, jumlah kelas kata yang ada bergantung pada sebanyak apa sebuah kata di bahasa Indonesia dipasangkan dengan kata bahasa Inggris yang berbeda dan memiliki makna pada *sense tagged english corpus*. Jumlah kelas ini dapat bergantung pada seberapa akurasi *alignment* kata yang dilakukan pada proses sebelumnya.

Pada sebaran jumlah *instance* di dalam kelas-kelasnya, kata "kehadiran" memiliki jumlah kelas *attendance* (19 buah), *presence* (64 buah), dan *existence* 1 buah. Perbandingan kedua kelas dengan *instance* terbanyak tersebut adalah 19:64 yaitu 0.29. Kedua *sense* tersebut memiliki makna yang kurang lebih menyatakan

sebuah *state* dimana seseorang datang/hadir. Sementara itu, kelas kata "rumahnya" memiliki jumlah *instance* sebanyak 32 buah pada kelas "house", dan 19 buah pada kelas "home". Perbandingan kedua kelas pada kata "rumahnya" tersebut adalah 0.59, yang mana rasio tersebut yang lebih besar dari rasio kelas kata "kehadiran" (0.29). Tabel 5.7 menunjukan makna kata yang dipindahkan berdasarkan *sampling* berdasarkan sebaran *instance* dalam kelas.

Tabel 5.7: Evaluasi *Sense Transfering* Berdasarkan Sebaran Kelas

Kata (*sense key*)	**Makna**	**Kalimat**	**Jumlah**
Kehadiran (attendance%1:04:00::)	*the act of being present (at a meeting or event etc.)*	...tingkat **kehadiran** guru di sekolah...	19
Kehadiran (presence%1:09:00::)	*the impression that something is present*	...berkurangnya **kehadiran** pria dewasa...	64
rumahnya (house%1:14:02::)	*an official assembly having legislative powers*	...kebakaran yang melanda **rumahnya**...	32
rumahnya (home%1:06:00::)	*Housing that someone is living in*	...ia pulang ke **rumahnya** pada sabtu...	19

Kedua makna pada kata "kehadiran" memiliki makna yang relatif dekat dan sesuai dalam konteks kalimat kata tersebut muncul. Namun demikian, *sense key* pada kata rumah yaitu "house%1:14:02::" memiliki makna yang salah, dimana sense key yang lebih tepat semestinya adalah house%1:06:01:: dengan makna "*a building in which something is sheltered or located*". Kesalahan makna kata yang dipindahkan tersebut disebabkan karena kata *house* pada korpus inggris diberikan tag 'house%1:14:02::'. Kesalahan ini seperti yang sudah dijelaskan pada analisis *tagging* IMS pada sub-bab Pembuatan *Sense Tagged Corpus* Bahasa Inggris.

Kelompok *sampling* lain adalah makna yang akan dilihat pada kata dengan bentuk morfologi yang berbeda. Kata "makan", "makanan", dan "memakan" merupakan kata yang mewakili kasus bentuk morfologi dalam bentuk lemma maupun berimbuhan. Pada kata "makan", *sense key* yang diterima dari hasil *transfer* adalah eat%2:34:00:: yang memiliki makna "*take in solid food*". Kata "makanan" pada kalimat-kalimat yang ada diberikan *sense key* food%1:03:00::

yang diartikan sebagai "*any substance that can be metabolized by an animal to give energy and build tissue*". Kata "memakan" sendiri memiliki beberapa *sense key* seperti consume%2:34:02::, eat%2:34:00::, dan feed%1:13:00::. Dari *sense key* yang didapat tersebut, consume%2:34:02::("spend extravagantly") bukan merupakan *sense* yang tepat (seharusnya memiliki makna mengonsumsi makanan), dan feed%1:13:00:: ("*food for domestic livestock*") yang semestinya adalah "memberikan makanan".

Terdapat beberapa kata-kata yang dipilih secara manual untuk dihitung akurasi dari kesesuaian *tag* makna yang didapat. Kriteria pemilihan kata-kata sebagai *sample* ini yaitu memiliki pasangan kata dalam Bahasa Inggris dan makna kata lebih dari satu. Evaluasi ini akan menghitung seberapa banyak kelas-kelas makna kata yang benar dari hasil *transfering* pada kata-kata pilihan tersebut. Hasil akurasi dari jumlah makna kata yang tepat dapat dilihat pada tabel 5.8.

Kata	**Jumlah *sense key***	***sense key***	**Jumlah *sense key* yang benar**	**Akurasi**
halaman	3	page%1:10:00::, yard%1:23:00::, courtyard%1:06:00::	2	0.67
coklat	3	cocoa%1:13:02, cacao%1:20:00::, brown%5:00:00:chromatic:00	3	1.0
debu	2	dust%1:27:00::, plume%1:06:00::	1	0.5
batasan	7	limit%1:07:00::, restriction%1:09:00::, restraint%1:04:00::, limitation%1:23:00::, definition%1:10:00::, constraint%1:26:00::, limit%2:30:01::	5	0.71

lingkungan	7	environment%1:26:00::, environmental%3:01:01::, neighborhood%1:14:00::, environmentally%4:02:00::, sphere%1:15:00::, surroundings%1:15:00::, outside%5:00:00:external:00	5	0.71
hakim	3	judge%1:18:01::, justice%1:04:00::, magistrate%1:18:00::	2	0.67
hati	3	heart%1:09:00::, liver%1:08:00::, mind%1:09:00::	2	0.67
memecahkan	4	solve%2:31:00::, resolve%2:31:01::, break%2:30:03::, split%2:38:00::	3	0.75
Rerata	-	-	-	**0.73**

Tabel 5.8: Evaluasi *sense transfering* akurasi kelas makna kata

Berdasarkan tabel di atas, rerata akurasi yang didapatkan pada kata-kata tersebut adalah 0.73. Terdapat beberapa makna hasil *transfer* yang tidak sesuai dengan kata pasangannya seperti misalnya "plume%1:06:00::" dengan kata "debu" yang memiliki makna "*a feather or cluster of feathers worn as an ornament*, atau *sense key* "limit%2:30:01::" dengan makna "*place limits on (extent or access)*" pada kata "batasan". *Sense key* "limit%2:30:01::" tersebut dianggap tidak sesuai dengan kata "batasan" karena makna yang dikandung adalah "memberi batas" dan bukan "batasan" itu sendiri.

Pada hasil tabel tersebut, dipilih beberapa kata-kata yang ambigu (memiliki makna kata yang tidak mirip) secara manual untuk dievaluasi pada tahap berikutnya. Pada tahap ini, evaluasi dilakukan anotator dengan menghitung berapa jumlah makna hasil *transfer* pada kata tersebut, yang sesuai dengan kalimat dimana kata itu muncul. Makna kata yang tidak sesuai dari hasil evaluasi sebelumnya tidak dimasukkan ke dalam proses ini. Hasil evaluasi dapat dilihat pada tabel 5.9.

Tabel 5.9: Evaluasi *sense transfering* berdasarkan jumlah kesesuaian dengan kalimat

Kata dan *sense key*	**Jumlah benar**	**Jumlah kalimat**	**Akurasi**
halaman (page%1:10:00::)	29	41	0.70
halaman (courtyard%1:06:00::)	3	3	1.0
coklat (cocoa%1:13:02::)	6	10	0.6
coklat (brown%5:00:00:chromatic:00)	3	4	0.75
coklat (cacao%1:20:00::)	4	5	0.8
hati (heart%1:09:00::)	166	170	0.97
hati (liver%1:08:00::)	44	44	1.0
memecahkan (solve%2:31:00::)	12	13	0.92
memecahkan (resolve%2:31:01::)	4	7	0.57
memecahkan (split%2:38:00::)	1	1	1.0
Rerata Akurasi	-	-	**0.848**

Pada hasil evaluasi tersebut, ketidaksesuaian *sense* "page%1:10:00::" (halaman buku) pada kata "halaman", terjadi pada kalimat yang mengandung makna "halaman *web*" dan bukan halaman buku. Kesalahan pada kata "hati" juga disebabkan karena "*heart*" diberikan *tag* "hati (perasaan)" dan pada beberapa kasus, "*heart*" yang dimaksud adalah "*heart* (jantung)".

Berdasarkan berbagai hasil evaluasi di atas, makna kata yang tidak tepat merupakan kesalahan dari baik itu variasi *alignment* suatu kata yang dipasangkan dengan kata lainnya, ataupun juga hasil *tagging* model IMS yang yang tidak tepat. Pada kesalahan hasil *alignment* kasus yang terjadi adalah ketika suatu kata dipasangkan dengan kata lain yang tidak tepat sehingga berpengaruh pada makna kata yang dipindahkan. Kesalahan *tagging* pada IMS juga mengakibatkan kesalahan *sense key* pada kata Bahasa Indonesia karena pada *sense tagged corpus* sudah mengalami kesalahan.

5.4 WSD Bahasa Indonesia

Untuk melihat seberapa baik performa sistem WSD dengan menggunakan *sense tagged corpus* hasil dari penelitian, terdapat kata-kata yang dipilih secara manual sebagai sampling dari *target word* yang akan dievaluasi berdasarkan nilai F-score dari hasil rata-rata *cross validation*. Kata yang dipilih merupakan *instance* yang memiliki pasangan kata lebih dari satu dalam bahasa Inggris dan mempunyai makna yang berbeda dari hasil *sense transfering*. *Target word* yang dipilih tersebut memiliki kriteria bahwa pasangan bahasa inggris kata tersebut lebih dari satu ataupun juga pasangan bahasa inggrisnya memiliki makna yang tidak dekat (ambigu). Fitur yang dilakukan percobaan pada penelitian ini adalah F1(*bag of words*), F2 (*word embedding*), F3 (*pos-tag*), F4 (*pos tagging* dan *bag of words*). Hasil evaluasi dapat dilihat pada tabel akurasi 5.10 berikut.

Tabel 5.10: Evaluasi sistem WSD

Kata	**Baseline**	**F1**	**F2**	**F3**	**F4**
memecahkan	0.5	0.51	0.62	0.48	0.51
menolak	0.6	0.63	0.6	0.76	0.74
obat	0.49	0.58	0.62	0.56	0.63
lingkungan	0.54	0.51	0.42	0.7	0.66
halaman	0.93	0.9	0.88	0.83	0.88
kehadiran	0.67	0.93	0.8	0.72	0.88
hati	0.72	0.88	0.82	0.75	0.89
coklat	0.33	0.61	0.48	0.34	0.67
berat	0.53	0.5	0.46	0.64	0.66
jalan	0.66	0.72	0.74	0.69	0.72

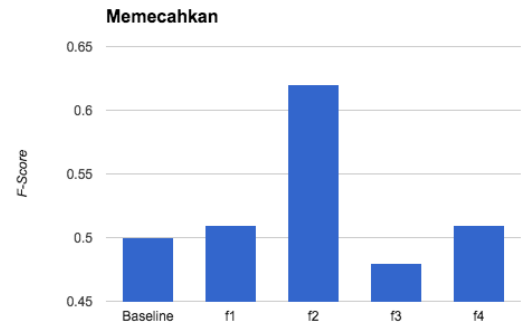
Pada hasil akurasi (F-Score) tabel tersebut, dapat terlihat bahwa pada fitur F1, tujuh dari sepuluh (70%) buah kata memiliki nilai performa di atas baseline, tujuh dari sepuluh kata (70%) mengungguli baseline pada fitur F2, delapan dari sepuluh kata (80%) mengungguli baseline pada fitur F3, dan sembilan dari sepuluh kata (90%) mengungguli baseline pada fitur F4. Rerata performa dari hasil tersebut dapat dilihat pada tabel 5.11.

Tabel 5.11: Performa rerata sistem WSD Bahasa Indonesia

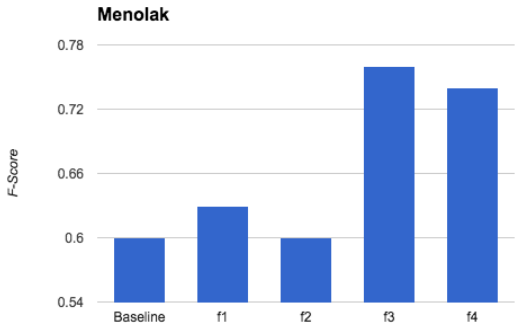
Baseline	**F1**	**F2**	**F3**	**F4**
0.597	0.677	0.644	0.647	0.724

Berdasarkan rata-rata dari nilai F-Score tersebut, dapat dilihat bahwa untuk kesepuluh data *sampling* yang digunakan, rerata dari F-Score semua fitur mengungguli sistem baseline.

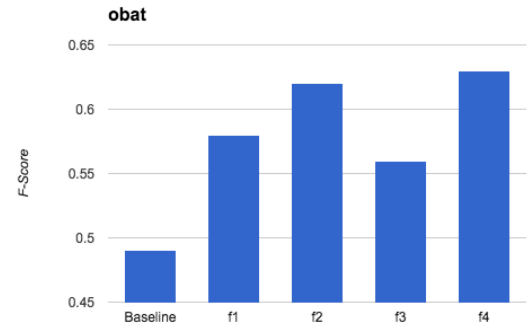
Grafik perbandingan dari fitur-fitur tersebut untuk setiap kata dapat dilihat pada gambar 5.1.



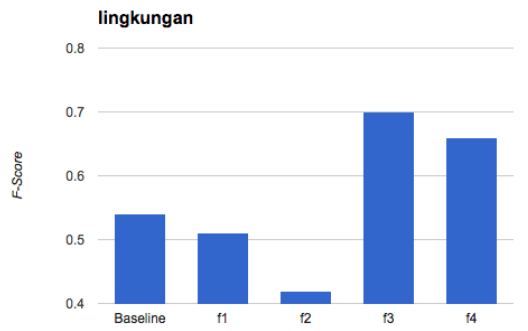
(a) memecahkan



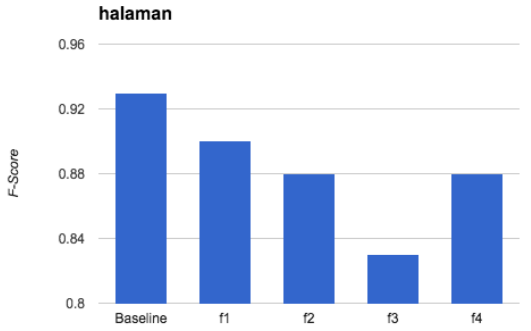
(b) menolak



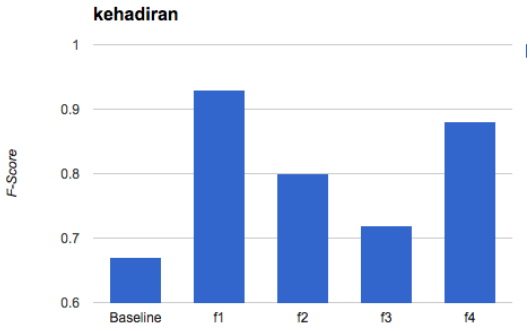
(c) obat



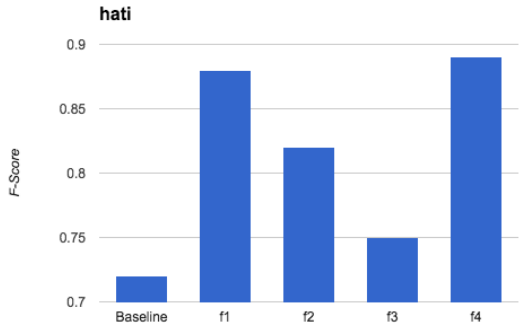
(d) lingkungan



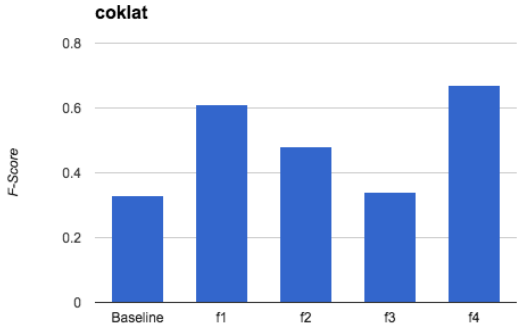
(e) halaman



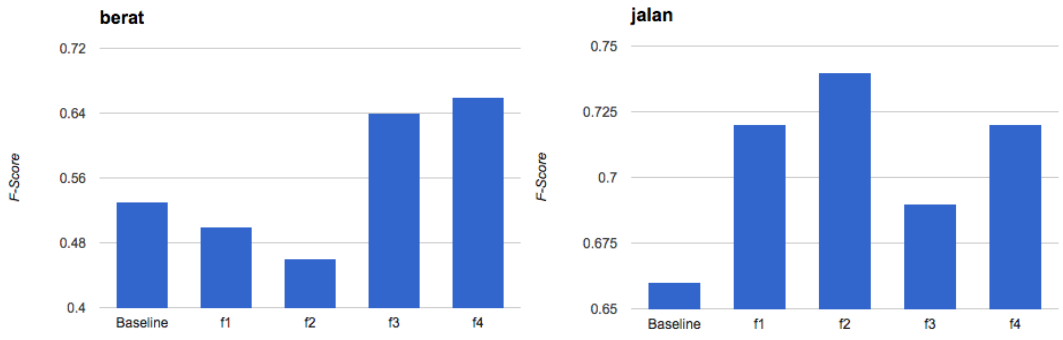
(f) kehadiran



(g) hati



(h) coklat



(i) berat **(j)** jalan

Gambar 5.1: Grafik Performa Fitur WSD Bahasa Indonesia

Berdasarkan sepuluh grafik di atas, dapat kita lihat bahwa sembilan dari sepuluh *sample* kata yang dicoba, hanya satu kata yang tidak pernah memiliki performa lebih baik baseline yaitu kata "halaman". Sembilan kata lainnya selalu memiliki performa di atas baseline untuk minimal satu dari keempat fitur yang dicoba. Pada tujuh buah *sample*, fitur F2 memiliki akurasi di bawah F1 walaupun sudah menggunakan *word embedding*. Pada lima buah *sample* kata, F4 yang merupakan fitur gabungan F1 dan F3 memiliki performa yang lebih baik dari fitur F1 saja. Penggunaan berbagai skenario fitur tersebut menunjukan bahwa fitur F1 atau F4 memiliki performa minimal sama atau di atas baseline pada sembilan dari sepuluh *sample* (90%), hal ini menunjukkan bahwa *bag of words* ataupun kombinasinya dengan POS Tag dapat mengungguli baseline pada *sampling* ini. Jumlah persentase pada *sampling* dengan fitur F2 yang mengungguli baseline adalah 70%. Performa F2 yang lebih rendah dari F4 pada rerata F-Score dapat disebabkan karena domain yang berbeda antara *training data* pada saat membentuk model *word embedding* (perbedaan pada gaya bahasa pada Wikipedia dan korpus identik). Namun demikian, persentase terbesar dari *sampling* sepuluh kata di atas diperoleh pada fitur F4 yang memiliki performa lebih baik dari baseline pada sembilan kasus dari sepuluh kata tersebut (90%).

BAB 6
PENUTUP

6.1 Kesimpulan

Data dan *resource* berupa *sense tagged corpus* yang terbatas pada bahasa Indonesia merupakan penghambat dari penelitian WSD di Bahasa Indonesia. Untuk mengatasi masalah tersebut, salah satu pendekatan WSI berupa *cross lingual* dapat dimanfaatkan untuk *transfering knowledge* dari salah satu bahasa dengan data yang lebih banyak yaitu bahasa Inggris.

Konsep pendekatan *cross lingual sense transfering* ini dapat menghasilkan *sense tagged corpus* bahasa Indonesia dengan memindahkan makna kata dari korpus bahasa Inggris ke kata-kata yang menjadi pasangannya (*translation* dari kata tersebut) di dalam Bahasa Indonesia. Metode yang digunakan untuk membangun *sense tagged corpus* tersebut meliputi *tagging* pada korpus Bahasa Inggris dengan menggunakan *tool* IMS (Zhong dan Ng, 2010), melakukan *alignment* kata pada korpus identik (Inggris-Indonesia) dengan Giza++ (Och dan Ney, 2003), dan *sense transfering* untuk memindahkan makna kata tersebut. *Sense tagged corpus* Bahasa Indonesia yang merupakan hasil dari proses tersebut memiliki kualitas cukup baik walaupun masih terdapat makna kata yang tidak tepat karena kesalahan pada proses *tagging* korpus bahasa Inggris ataupun *alignment*.

Pada penelitian yang dilakukan, sistem WSD dibuat dengan *classifier* SVM dan dicoba dengan empat skenario fitur, yaitu *bag of words* (F1), *word embedding* (F2), POS Tag (F3), dan gabungan antara F1 dengan F3 (F4). Sistem tersebut memiliki performa yang lebih baik dari baseline pada kebanyakan kasus untuk fitur tertentu. Fitur dengan persentase terbaik pada penelitian ini adalah gabungan dari fitur *bag of words* dan POS Tag yang mengungguli baseline pada 90% *sample* kata percobaan dengan nilai rerata F-Score terbesar yaitu 72.4%.

6.2 Saran

Setelah melakukan percobaan dan melakukan analisis hasil dari penelitian ini, terdapat beberapa saran untuk penelitian selanjutnya diantaranya sebagai berikut.

1. Penggunaan kualitas dari korpus paralel dapat mempengaruhi seberapa baik hasil *sense transfering* yang dilakukan. Pada korpus identik yang penelitian

ini gunakan dengan jumlah kalimat sebanyak 88.919 buah, terdapat beberapa kalimat yang terpotong maupun *comparable* (tidak sepenuhnya paralel), sehingga rawan menimbulkan kesalahan *alignment*.

2. Fitur yang digunakan dalam sistem WSD bahasa Indonesia masih dapat ditambahkan dengan berbagai fitur lainnya seperti Collocation, *text rank*, *dependancy parser*, dan lain-lain.

3. *Classifier* pada sistem WSD juga dapat diganti dengan yang lain seperti misalnya Naive Bayes, Multilayer Perceptron, dan lain-lain. Arsitektur juga dapat dicoba untuk menggunakan pendekatan *deep learning* pada penelitian selanjutnya.

4. Proses *word alignment* sebaiknya dilakukan dengan *tool* SMT yang mempunyai model lebih baru seperti misalnya berkeley aligner. Selain *tool*, teknik hybrid seperti misalnya menggabungkan *alignment* dari *tool* yang berbeda juga mungkin dapat dicoba untuk meningkatkan kualitas dari *word alignment* yang dihasilkan.

DAFTAR REFERENSI

Denkowski, M. (2009). A survey of techniques for unsupervised word sense induction. *Language & Statistics II Literature Review*, pages 1–18.

Dinakaramani, A., Rashel, F., Luthfi, A., dan Manurung, R. (2014). Designing an indonesian part of speech tagset and manually tagged indonesian corpus. In *Asian Language Processing (IALP), 2014 International Conference on*, pages 66–69. IEEE.

Fradkin, D. dan Muchnik, I. (2006). Support vector machines for classification. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 70:13–20.

Gale, W. A., Church, K. W., dan Yarowsky, D. (1992). One sense per discourse. In *Proceedings of the workshop on Speech and Natural Language*, pages 233–237. Association for Computational Linguistics.

Larasati, S. D. (2012). Identic corpus: Morphologically enriched indonesian-english parallel corpus. In *LREC*, pages 902–906.

Mihalcea, R. dan Pedersen, T. (2003). An evaluation exercise for word alignment. In *Proceedings of the HLT-NAACL 2003 Workshop on Building and using parallel texts: data driven machine translation and beyond-Volume 3*, pages 1–10. Association for Computational Linguistics.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., dan Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Nasiruddin, M. (2013). A state of the art of word sense induction: A way towards word sense disambiguation for under-resourced languages. *arXiv preprint arXiv:1310.1425*.

Och, F. J. dan Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Pantel, P. dan Lin, D. (2002). Discovering word senses from text. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 613–619. ACM.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., dan Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Rudnick, A. (2011). Towards cross-language word sense disambiguation for quechua. In *RANLP Student Research Workshop*, pages 133–138.

Septiantri, H. (2013). Word sense disambiguation (wsd) untuk bahasa indonesia menggunakan *Cross-Lingual* wsd dengan korpus paralel. unpublished thesis.

Sheeba, J., Vivekanandan, K., Sabitha, G., dan Padmavathi, P. (2013). Unsupervised hidden topic framework for extracting keywords (synonym, homonym, hyponymy and polysemy) and topics in meeting transcripts. In *Advances in Computing and Information Technology*, pages 299–307. Springer.

Taghipour, K. dan Ng, H. T. (2015). One million sense-tagged instances for word sense disambiguation and induction. In *CoNLL*, pages 338–344.

Tala, Z. (2003). A study of stemming effect on information retrieval in bahasa indonesia, theses. *Institute for Logic, Language and Computation, Universiteit van Amsterdam, The Nedherlands*.

Ting, K. M. (2011). Precision and recall. In *Encyclopedia of machine learning*, pages 781–781. Springer.

Zhong, Z. dan Ng, H. T. (2010). It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 System Demonstrations*, pages 78–83. Association for Computational Linguistics.

LAMPIRAN 1 : PANDUAN EVALUASI *WORD ALIGNMENT*

Pengantar

1. Evaluasi ini dilakukan untuk mengetahui seberapa baik performa alignment yang dilakukan oleh tool Giza++ pada korpus identik. Korpus identik ini merupakan kumpulan pasangan kalimat dalam dua bahasa yaitu Indonesia-Inggris.
2. Anotator (evaluator) pada task ini bertugas untuk memasangkan kata-kata dari bahasa Inggris-Indonesia pada pasangan kalimat yang diberikan.
3. Input yang diberikan adalah file yang berisi 200 pasang kalimat Indonesia-Inggris dengan format:

   Sentence pair#<*nomor*>
   <*Kalimat bahasa Indonesia*>
   <*Kalimat bahasa inggris*>
   Sentence pair#<*nomor+1*>
   <*Kalimat bahasa Indonesia*>
   <*Kalimat bahasa inggris*>
   …
   Sentence pair#200
   <*Kalimat bahasa Indonesia*>
   <*Kalimat bahasa inggris*>

   Dengan setiap kalimat memiliki format:
   Kalimat bahasa Indonesia

   kata1(indeks1) kata2(indeks2) … kataN(indeksN)

   Kalimat bahasa Inggris

   Kata1 ({ }) kata2 ({ }) kata3 ({ }) … kataM ({ })

4. Output merupakan file yang sama, dengan kata-kata pada bahasa Inggris sudah dipasangkan dengan nomor indeks kata bahasa Indonesia (Contoh pemasangan dapat dilihat pada bagian **proses pemasangan kata**)
5. Pasangan kalimat secara sederhana dapat dicontohkan dengan kalimat input berikut:

   Sentence pair#1
   Kamu(1) seharusnya(2) belajar(3) dari(4) hal(5) tersebut(6)
   NULL ({ }) You ({ }) should ({ }) have ({ }) learned ({ }) from ({ }) that ({ })

   Tugas anotator adalah memasangkan kata-kata dalam bahasa inggris dengan kata yang bersesuaian pada bahasa Indonesianya. Pada contoh kalimat tersebut, pasangan kata yang tepat adalah:
   - “You” dengan “Kamu”

- “should” dengan “seharusnya”
- “learned” dengan “belajar”
- “from” dengan “dari”
- “that” dengan “hal tersebut”

Proses Pemasangan Kata

1. Diberikan pasangan kalimat dalam bahasa Inggris dan bahasa Indonesia sebanyak 200 buah pasangan:

Contoh **input** Pasangan 1:

```
Sentence pair#1
Lalu(1) ia(2) mengalami(3) kedamaian(4) yang(5) tidak(6) terlukiskan(7)
NULL ({ }) He ({ }) then ({ }) experienced ({ }) an ({ }) indescribable ({ }) peace ({ })
```

Cara membaca:

Indeks	1	2	3	4	5	6	7
Kata	Lalu	ia	mengalami	kedamaian	yang	tidak	terlukiskan

2. Dari pasangan kalimat tersebut lakukan pemasangan **satu kata dengan kata lain** dengan cara:

2.1. Berikan angka indeks yang sesuai pada kata-kata dalam bahasa inggris yang bersesuaian (Kata yang bersesuaian tidak harus *direct translation* tetapi bisa yang maknanya serupa).

Contoh pada pasangan 1:

- Kata “**ia**” berpasangan dengan “**he**”, maka **He ({ })** diisi dengan angka 2(ia)
- Sehingga menjadi **He ({ 2 })**
- Kata “**indescribable**” berpasangan dengan “**tidak**” dan “**terlukiskan**”, maka **indescribable ({ })** diisi dengan angka **6(tidak)** dan **7(terlukiskan)**
- Sehingga menjadi **indescribable ({ 6 7 })**

2.2. Berikan angka indeks pada token **NULL**, pada kata-kata bahasa Indonesia yang tidak memiliki pasangan.

Contoh pada pasangan 1:

- Kata “**yang**” pada kalimat tersebut tidak memiliki pasangan, sehingga diletakan pada token **NULL ({ })**
- Sehingga menjadi **NULL ({ 5 })**

2.3. Hasil akhir merupakan kalimat bahasa inggris yang sudah dipasangkan dengan angka indeks tersebut.

Contoh hasil output:

```
Sentence pair#1
Lalu(1) ia(2) mengalami(3) kedamaian(4) yang(5) tidak(6) terlukiskan(7)
NULL ({ 5 }) He ({ 2 }) then ({ 1 }) experienced ({ 3 }) an ({ }) indescribable ({ 6 7 })
peace ({ 4 })
```

3. **Bila terdapat kebingungan dalam memasangkan kata**, *annotator* dapat membuka kamus Indonesia-English untuk mencari *vocabulary* yang dibutuhkan.
4. **Bila terdapat kalimat yang terpotong atau tidak selaras dengan kalimat pasangannya**, laporkan nomor kalimat tersebut untuk dapat diganti dengan kalimat lain.

Contoh kasus yang mungkin terjadi:

1. Terdapat satu kata bahasa Inggris yang mempunyai pasangan dengan lebih dari satu kata dalam bahasa Indonesia seperti:
    a. **Hospital** dengan **rumah sakit**
    b. **Smaller** dengan **lebih kecil**

   Contoh kasus dalam pasangan kalimat:

```
Tidak(1) seorang(2) pun(3) dari(4) klub(5) tersebut(6) yang(7) segera(8) memberi(9)
komentar(10)
NULL ({ 4 7 9 6 }) Nobody ({ 1 2 3 }) at ({ }) the ({  }) club ({ 5 }) was ({ }) immediately ({
8 }) available ({ }) for ({ }) comment ({ 10 })
```

   Kata "**Tidak**", "**seorang**", dan "**pun**" dipasangkan dengan kata "**Nobody**".

2. Terdapat satu kata bahasa Indonesia yang mempunyai pasangan dengan lebih dari satu kata dalam bahasa Inggris seperti:
    a. **Menyukainya** dengan **like him**
    b. **Menyerah** dengan **give up**

   Contoh kasus dalam pasangan kalimat:

```
Akhirnya(1) ani(2) menyerah(3) dalam(4) lomba(5) itu(6)
NULL ({ }) ani ({ 2 }) finally ({ 1 }) gives ({ 3 }) up ({ 3 }) on ({ 4 }) that ({ 6 }) competition
({ 5 })
```

3. Terdapat kata-kata yang tidak memiliki pasangan pada kalimat bersangkutan seperti:

   Contoh pada pasangan kalimat:

```
tahun(1) yang(2) lalu(3) hanya(4) sekitar(5) 4(6) sekian(7) persen(8)
```

NULL ({ 2 4 5 7 }) Last ({ 3 }) year ({ 1 }) it ({ }) expanded ({ }) 4 ({ 6 }) percent ({ 8 })

Kata “**yang**”, “**hanya**”, “**sekitar**”, dan “**sekian**” dipasangkan dengan **NULL** karena tidak ada kata yang bersesuaian pada kalimat bahasa Inggris diatas.

4. Terdapat kata-kata yang merupakan terjemahan secara implisit, seperti

Pekerjaan(1) yang(2) sangat(3) menyenangkan(4) NULL ({ 2 }) awesome ({ 3 4 }) job ({ 1 })

5. Kasus dimana terdapat panggilan/singkatan, seperti:

Susilo(1) Bambang(2) Yudhoyono(3) turun(4) dari(5) jabatan(6) NULL ({ 4 5 6 }) SBY ({ 1 2 3 }) is ({ }) retired ({ 4 5 6 })

Atau contoh lain

Susilo(1) Bambang(2) Yudhoyono(3) (SBY)(4) turun(5) dari(6) jabatan(7) NULL ({ 4 5 6 }) SBY ({ 1 2 3 4 }) is ({ }) retired ({ 5 6 7 })

6. Kasus dimana terdapat tanda baca pada kalimat, seperti:

Dia(1) gagal(2) menuntaskannya!(3) NULL ({ }) He ({ 1 }) is ({ }) failed!({ 2 }) to ({ }) finish ({ 3 }) it ({ 3 })

Pada kasus masih terdapat tanda baca di kata-kata dalam kalimat, **abaikan** tanda baca tersebut.