PEMBANGUNAN KAMUS INDONESIA-JEPANG DARI KORPUS DWIBAHASA DENGAN SVM

Laporan Tugas Akhir I

Disusun sebagai syarat kelulusan mata kuliah
IF4091/Tugas Akhir I dan Seminar

Oleh
MUHAMMAD NASSIRUDIN
NIM : 13511044

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
Januari 2015

PEMBANGUNAN KAMUS INDONESIA-JEPANG DARI KORPUS DWIBAHASA DENGAN SVM

Laporan Tugas Akhir I

Oleh

MUHAMMAD NASSIRUDIN

NIM : 13511044

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

Bandung, 16 Januari 2015

Mengetahui,

Pembimbing,

Dr. Eng. Ayu Purwarianti, S.T.

NIP. 197701272008012011

DAFTAR ISI

DAFTAR LAMPIRAN

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR SINGKATAN DAN ISTILAH

Singkatan/Istilah	Deskripsi	Hal.
ATR	*Automatic Term Recognition*.	1
NLP	*Natural Language Processing*.	1
PoS	*Part of Speech*.	2
SVM	*Support Vector Machine*.	3
atribut	Ukuran yang digunakan untuk merepresentasikan sebuah *instance*.	10
fitur	Aspek yang dilihat untuk membedakan satu *instance* dari yang lain.	11
hipotesis	Sebuah fungsi yang berusaha mengestimasi konsep/fungsi target sedekat-dekatnya.	11
instance	Satuan individu/anggota dari domain yang dipelajari.	10
korpus	Sekumpulan sumber teks atau suara yang dapat dibaca oleh mesin yang ditujukan untuk penelitian pemrosesan bahasa.	1
***n*-gram**	Barisan *n* kata berurutan dalam sebuah teks.	2
stop word	Kata yang sering muncul dalam dokumen, tetapi tidak ada kaitan erat dengan topik yang dibahas seperti kata hubung dan kata sandang.	24

BAB I

PENDAHULUAN

I.1 Latar Belakang

Bertambah pesatnya perkembangan kerja sama antara Indonesia dan Jepang membuat semakin sering terjadinya interaksi antarbudaya dari masing-masing negara. Salah satu aspek yang tentunya juga terkena dampak adalah bahasa. Perbedaan bahasa merupakan sebuah kekayaan dunia, namun hal tersebut justru menjadi masalah ketika sebuah domain tertentu sudah terlibat. Bahasa yang berbeda menggunakan istilah yang berbeda untuk menyatakan satu hal yang sama, seperti コンピューター (konpyu-ta-) dalam bahasa Jepang yang berarti komputer dalam bahasa Indonesia. Perbedaan tersebut menjadi masalah karena menyebabkan sulitnya memastikan apakah orang yang berbahasa Indonesia akan memikirkan hal yang sama dengan orang yang berbahasa Jepang ketika disebutkan sebuah istilah dalam sebuah domain. Contohnya, kata "motor" di Indonesia identik dengan kendaraan yang bermesin dan beroda dua, sedangkan salah satu kata padanannya di Jepang, "モーター" (mo-ta-), identik dengan mesin yang menggerakkan sesuatu. Contoh nyata dari masalah tersebut adalah pembuatan buku panduan dari sebuah produk elektronik yang tentunya harus menggunakan istilah yang dapat dimengerti calon pembeli dari berbagai macam negara.

Salah satu solusinya adalah pembuatan sebuah kamus Indonesia-Jepang spesifik domain. Solusi seperti ini pernah dilakukan sebelumnya oleh Lefever, Macken, dan Hoste (2009). Lefever dkk. (2009) membangun kamus multibahasa berdomain otomotif untuk 4 bahasa, yaitu Prancis, Inggris, Italia, dan Belanda. Pembangunan kamus tersebut ditujukan untuk meningkatkan konsistensi istilah teknis dalam sebuah perusahaan otomotif Prancis. Pembangunan kamus dilakukan menggunakan *Automatic Term Recognition* (ATR) yang merupakan salah satu topik dari *Natural Language Processing* (NLP). ATR digunakan untuk mengekstrak istilah-istilah yang muncul bersamaan dari masing-masing bahasa secara otomatis melalui korpus. Pembangunan kamus kemudian dilanjutkan dengan menyaring pasangan isti-

lah tersebut berdasarkan pengukuran statistik.

Sampai saat ini, dikenal 2 jenis korpus dalam ATR untuk pembuatan kamus dwibahasa, yaitu (Koehn, 2010) korpus paralel dan korpus *comparable*. Dua buah korpus dikatakan paralel jika yang satu merupakan translasi langsung dari korpus yang lain sementara dua buah korpus dikatakan *comparable* jika yang satu bukan translasi langsung dari korpus yang lain, namun masih setopik. Ekstraksi istilah dari korpus paralel lebih mudah dibandingkan dari korpus *comparable*, namun lebih sulit didapatkan. Sejauh ini, korpus paralel untuk pasangan bahasa Indonesia-Jepang dapat ditemukan pada buku-buku manual produk elektronik, sementara korpus *comparable* dapat ditemukan dalam artikel-artikel Wikipedia[1].

Teknik ATR dapat dilakukan dengan 2 macam pendekatan (Lefever dkk., 2009), yaitu pendekatan linguistik dan pendekatan statistik. Pendekatan linguistik memanfaatkan informasi linguistik, seperti *Part of Speech* (PoS) *tag*, frasa, dan pola sintaksis. Teknik yang menggunakan pendekatan linguistik antara lain pemanfaatan transliterasi untuk menangkap kata-kata yang memiliki akar bahasa yang sama (Tsuji, Daille, & Kageura, 2002) dan pemanfaatan kamus dwibahasa yang sudah ada untuk menghitung peluang apakah sebuah pasangan kata adalah pasangan translasi (Jagarlamudi & Daumé III, 2010). Pendekatan statistik dilakukan dengan memilih sekelompok kata yang berdekatan (*n-gram*) dan menggunakan perhitungan statistik untuk menyaring pasangan kata yang ekuivalen. Teknik yang menggunakan pendekatan statistik pernah dilakukan oleh Daille dan Morin (2005) dengan melakukan ekstraksi SWT (*single-word term*) dan MWT (*multi-word term*) kemudian menggunakan pengukuran *similarity vector* untuk melakukan *alignment* kata per kata pada MWT. Selain itu, Lefever dkk. (2009) juga menggunakan pendekatan statistik dengan melakukan pengenalan istilah melalui informasi frekuensi dan frekuensi *n-gram* dari korpus.

Pada praktiknya, kedua pendekatan tersebut tidaklah terpisah antara satu dengan yang lain. Terdapat beberapa penelitian terkait ATR yang mencoba menggabungkan fitur linguistik dan fitur statistik (pendekatan hibrid). Salah satunya adalah pe-

[1] `https://www.wikipedia.org/`

kerjaan yang dilakukan oleh Aker, Paramita, dan Gaizauskas (2013). Teknik yang dipakai terbilang cukup baru, yaitu dengan memandang ATR sebagai masalah klasifikasi: apakah setiap pasangan istilah dari masing-masing bahasa adalah pasangan translasi atau tidak. Klasifikasi sendiri dapat dilakukan secara otomatis oleh mesin yang merupakan salah satu topik dalam pembelajaran mesin. Salah satu teknik klasifikasi yang menjadi *state-of-the-art* dalam pembelajaran mesin saat ini adalah *Support Vector Machine* (SVM) (Cristianini & Shawe-Taylor, 2000).

Pembuatan kamus Indonesia-Jepang pernah dilakukan sebelumnya oleh Limanthie (2014). Teknik yang digunakannya adalah dengan menghitung frekuensi pasangan kata dari setiap korpus. Limanthie (2014) hanya mengambil kata benda sebagai kandidat istilah yang akan dipasangkan dan belum membuahkan hasil yang memuaskan. Tugas akhir ini berusaha membuat kamus Indonesia-Jepang yang memiliki kinerja yang lebih baik. Caranya dengan mengadopsi pendekatan yang dilakukan oleh Aker dkk. (2013) yang disesuaikan dengan ciri-ciri pasangan bahasa Indonesia-Jepang.

I.2 Rumusan Masalah

Pembuatan kamus dwibahasa Indonesia-Jepang spesifik domain yang pernah dilakukan oleh Limanthie (2014) hanya menggunakan kamus dwibahasa sebagai fitur linguistiknya dan masih dibangun secara manual. Selain itu, teknik tersebut masih bergantung pada kualitas korpus yang bagus dan kinerjanya belum memuaskan. Di sisi lain, masih sedikitnya penelitian yang terkait juga merupakan sebuah masalah tersendiri yang menyebabkan bahasa Indonesia masih dalam status *under-resource*. Oleh karena itu, perlu dilakukan sebuah penelitian yang melibatkan lebih banyak aspek linguistik dari bahasa Indonesia dengan melihat kualitas korpus yang terbatas. Penelitian dilakukan dengan cara berikut:

1. memaksimalkan penggunaan aspek linguistik;
2. menggunakan teknik yang tidak bergantung pada jenis korpus; dan
3. memanfaatkan pembelajaran mesin agar sistem dapat berkembang secara otomatis.

I.3 Tujuan

Terdapat 2 hal yang ingin dicapai dalam tugas akhir ini sebagai berikut.

1. Membuat sebuah sistem pembangunan kamus dwibahasa yang menggabungkan fitur linguistik dan fitur statistik tanpa bergantung pada jenis korpus yang dipakai.
2. Membuat sebuah kamus Indonesia-Jepang dalam domain *computer science* dengan presisi yang lebih baik dari kamus sebelumnya menggunakan pembelajaran mesin dengan algoritma klasifikasi biner SVM.

I.4 Batasan Masalah

Domain yang dilingkupi dalam tugas akhir ini hanya domain *computer science*.

I.5 Metodologi

Pengerjaan tugas akhir ini dibagi dalam beberapa tahap sebagai berikut.

1. Perancangan desain solusi

   Pada tahap ini, dirancang sebuah sistem pembangunan kamus dwibahasa yang merupakan hasil dari studi literatur. Tahap-tahap berikutnya akan mengacu kepada sistem yang dibangun pada tahap ini.
2. Pengumpulan kakas dan data

   Pada tahap ini, dikumpulkan berbagai data yang dibutuhkan dalam pembangunan kamus. Data yang dibutuhkan adalah (1) korpus dalam bahasa Indonesia dan Jepang, (2) kamus latih, (3) data latih, dan (4) data uji. Selain itu, dikumpulkan juga kakas-kakas pemrosesan bahasa alami untuk teks seperti *statistical machine translation* serta kakas untuk pembelajaran.
3. Implementasi

   Pada tahap ini, semua data dan kakas yang telah dikumpulkan digunakan untuk mengimplementasikan model yang telah dibuat. Implementasi lebih menekankan pada optimasi pembelajaran melalui *tuning* parameter algoritma pembelajaran. Selain itu, dilakukan juga optimasi terhadap data latih yang telah dibuat pada tahap sebelumnya. Tahap ini menghasilkan sebuah model

pembelajaran yang dapat dipakai untuk membangun kamus.

4. Pengujian

   Pada tahap ini, dilakukan 2 macam pengujian. Yang pertama adalah pengujian tertutup, yaitu model pembelajaran diuji dengan data latih. Yang kedua adalah pengujian terbuka, yaitu model pembelajaran diuji dengan korpus yang tidak dipakai sebagai data latih. Pada akhir tahap ini didapatkan nilai presisi dari masing-masing pengujian.

I.6 Jadwal Pelaksanaan Tugas Akhir

Jadwal pelaksanaan tugas akhir per minggu diberikan pada Tabel I.1.

Tabel I.1. Jadwal pelaksanaan tugas akhir

	Des				Jan				Feb				Mar				Apr				Mei			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Perancangan Desain Solusi																								
Pembuatan arsitektur																								
Pengumpulan Kakas dan Bahan																								
Pengumpulan korpus																								
Pembuatan data latih																								
Pembuatan data uji																								
Pembuatan kamus latih																								
Pengumpulan kakas																								
Implementasi																								
Optimasi data latih																								
Pembelajaran mesin																								
Pengujian																								
Pengujian tertutup																								
Pengujian terbuka																								

BAB II

STUDI LITERATUR

II.1 Kamus Dwibahasa

Istilah "kamus dwibahasa" digunakan sebagai padanan istilah bahasa Inggris "*bilingual dictionary*". Menurut kamus Oxford *online*[2], definisi kata "*bilingual*" adalah teks atau aktivitas yang dituliskan atau dilakukan dalam dua bahasa, sedangkan kata "*dictionary*" didefinisikan sebagai sebuah buku atau sumber elektronik yang mendaftarkan kata-kata dari sebuah bahasa (biasanya terurut sesuai abjad) dan memberikan padanan katanya dalam bahasa yang lain.

Menurut KBBI (Kamus Besar Bahasa Indonesia) *online*[3], definisi dari istilah "kamus dwibahasa" adalah kamus yang memuat kata atau gabungan kata suatu bahasa yang disusun secara alfabetis dengan penjelasan makna dan contoh pemakaiannya di dalam bahasa lain yang menjadi bahasa sasaran. Dari ketiga definisi tersebut, dapat ditarik kesimpulan bahwa setidaknya terdapat 5 unsur utama yang ada di dalam sebuah kamus dwibahasa, yaitu

1. bahasa asal,
2. bahasa sasaran,
3. daftar kata atau frasa bahasa asal,
4. daftar kata atau frasa bahasa sasaran, dan
5. translasi antara kata atau frasa bahasa asal dengan bahasa sasaran.

Selain itu, Limanthie (2014) dalam tugas akhirnya menyebutkan, "Terdapat dua jenis kamus *bilingual*, yaitu kamus *unidirectional* dan kamus *bidirectional*. Kamus *unidirectional* hanya berisi terjemahan dari satu bahasa ke bahasa lain. Sedangkan kamus *bidirectional* berisi terjemahan dari satu bahasa ke bahasa lain dan sebaliknya." Contoh entri kamus dwibahasa diberikan pada Tabel II.1. Contoh kamus tersebut masuk ke dalam jenis *unidirectional* karena hanya menyediakan translasi dari bahasa Indonesia ke bahasa Jepang.

[2] `http://www.oxforddictionaries.com`
[3] `http://kbbi.web.id/`

Tabel II.1. Contoh entri kamus dwibahasa

No.	Bahasa Indonesia	Bahasa Jepang	Transliterasi
1 .	bahasa pemrograman	プログラミング言語	puroguramingu gengo
2 .	kamera digital	デジカメ	dejikame
3 .	kendaraan	車	kuruma
4 .	komputer	コンピューター	konpyu-ta-
5 .	mobil	車	kuruma
6 .	motor	オートバイ	o-tobai
7 .	motor	単車	tansha
8 .	motor	モーター	mo-ta-

II.2 Korpus

Korpus (atau korpora dalam bentuk jamak) dapat didefinisikan sebagai sekumpulan sumber teks atau suara yang dapat dibaca oleh mesin yang ditujukan untuk penelitian pemrosesan bahasa[4]. Dalam tugas akhir ini, hanya korpus berbentuk teks yang dipakai. Korpora yang berbentuk teks dapat diklasifikasikan ke dalam 2 jenis, yaitu (Koehn, 2010)

1. korpora paralel, yaitu korpus yang satu merupakan translasi langsung dari korpus yang lain (biasanya per kalimat) dan
2. korpora *comparable*, yaitu kedua korpus membahas topik/domain yang sama, tetapi korpus yang satu bukan translasi langsung dari korpus yang lain.

Korpora paralel lebih sulit dicari, namun sudah mulai banyak usaha yang dilakukan untuk membuatnya. Beberapa sumber korpora paralel yang dapat diakses saat ini adalah LDC[5] (*Linguistic Data Consortium*), Europarl[6], OPUS[7] (*the Open Parallel Corpus*), JRC-Acquis[8], dan DGT-TM[9] (DGT-*Translation Memory*). Bagaimanapun, kebanyakan dari sumber tersebut hanya menyediakan korpus untuk bahasa-bahasa yang ada di benua Eropa dan beberapa bahasa Asia yang banyak dipakai, seperti bahasa Arab dan bahasa Mandarin. Kendati demikian, OPUS sudah mulai menyediakan korpus paralel dengan kualitas bagus untuk bahasa Indonesia. Sela-

[4] http://www.oxforddictionaries.com
[5] https://www.ldc.upenn.edu/
[6] http://www.statmt.org/europarl/
[7] http://opus.lingfil.uu.se/
[8] https://ec.europa.eu/jrc/en/language-technologies/jrc-acquis
[9] https://ec.europa.eu/jrc/en/language-technologies/dgt-translation-memory

in itu, korpus paralel juga dapat dibuat secara manual dengan memanfaatkan buku manual dari peralatan elektronik, *crawling* dari laman *web*, *subtitle* film, novel terjemahan, dll. Pembuatan secara manual tentu saja membutuhkan usaha yang lebih besar. Contoh korpora paralel diberikan pada Tabel II.2.

Tabel II.2. Contoh potongan korpora *parallel* (Tiedemann, 2012)

Katakanlah: "Aku berlindung kepada Tuhan (yang memelihara dan menguasai) manusia. Raja manusia. Sembahan manusia. Dari kejahatan (bisikan) syaitan yang biasa bersembunyi, yang membisikkan (kejahatan) ke dalam dada manusia, dari (golongan) jin dan manusia."
言え，「ご加護を乞い願う，人間の主，人間の王，人間の神に。こっそりと忍び込み，囁く者の悪から。それが人間の胸に囁きかける，ジン（幽精）であろうと，人間であろうと。」

Korpora *comparable* lebih mudah ditemui karena sebuah topik yang sama akan dibicarakan oleh orang-orang dari berbagai belahan dunia. Contoh sumber yang menyediakan korpora *comparable* adalah koran, buku mengenai suatu domain, dan Wikipedia[10]. Wikipedia adalah sumber korpora *comparable* yang paling mudah didapatkan karena tersedia *online* dan mudah digunakan karena formatnya mudah dibaca oleh mesin (*machine-readable*). Namun, terdapat kasus artikel dengan topik yang sama dapat memiliki korelasi yang kecil karena untuk bahasa yang berbeda artikel tersebut dibuat oleh orang yang berbeda. Hal tersebut tentunya tidak bagus untuk kinerja ATR. Hal tersebut mungkin terjadi jika topiknya masih umum. Contoh korpora *comparable* diberikan pada Tabel II.3 yang diambil dari Wikipedia[11].

Dewasa ini, sudah mulai banyak penelitian yang berusaha mengaplikasikan ATR menggunakan korpora *comparable*. Hasilnya menyatakan bahwa ATR dapat bekerja dengan bagus menggunakan korpora *comparable* (dengan tingkat presisi mencapai 90%). Berdasarkan hal tersebut, penggunaan korpora *comparable* terbukti efektif untuk dipakai dalam ekstraksi istilah dwibahasa. Meskipun demikian, Koehn (2010) menyatakan bahwa korpora paralel jauh lebih baik untuk dipakai dalam ekstraksi *translational equivalence*.

[10] `https://www.wikipedia.org/`

[11] `http://id.wikipedia.org/wiki/Ilmu_komputer` dan `http://ja.wikipedia.org/wiki/`計算機科学

Tabel II.3. Contoh potongan korpora *comparable* dari Wikipedia

Ilmu komputer (bahasa Inggris: Computer Science), Secara umum diartikan sebagai ilmu yang mempelajari baik tentang komputasi, perangkat keras (hardware) maupun perangkat lunak (software). Ilmu komputer mencakup beragam topik yang berkaitan dengan komputer, mulai dari analisis abstrak algoritma sampai subyek yang lebih konkret seperti bahasa pemrograman, perangkat lunak, termasuk pe-rangkat keras. Sebagai suatu disiplin ilmu, Ilmu Komputer lebih menekankan pada pemrograman komputer, dan rekayasa perangkat lunak (software), sementara teknik komputer lebih cenderung berkaitan dengan hal-hal seperti perangkat keras komputer (hardware). Namun, kedua istilah tersebut sering disalah artikan oleh banyak orang.
計算機科学（けいさんきかがく、英: computer science）とは、情報と計算の理論的基礎、及びそのコンピュータ上への実装と応用に関する研究分野である[1][2][3]。計算機科学には様々な下位領域がある。コンピュータグラフィックスのように特定の処理に集中する領域もあれば、計算理論のように数学的な理論に関する領域もある。またある領域は計算の実装を試みることに集中している。例えば、プログラミング言語理論は計算を記述する手法に関する学問領域であり、プログラミングは特定のプログラミング言語を使って問題を解決する領域である。

II.3 Pembelajaran Mesin

Pembelajaran mesin ditujukan untuk membuat mesin dapat belajar melakukan suatu tugas tertentu secara otomatis. Dalam bukunya, Mitchell (1997) menyatakan, "Sebuah program komputer dikatakan belajar dari pengalaman E terhadap tugas T dan ukuran kinerja P jika kinerjanya terhadap tugas T, yang diukur dengan P, mengalami peningkatan dengan pengalaman E". Dalam mendesain sebuah sistem pembelajaran, beberapa hal yang perlu dilakukan antara lain: (Mitchell, 1997) (1) memilih pengalaman latih, (2) memilih fungsi target, (3) memilih representasi fungsi target, dan (4) memilih algoritma pembelajaran.

Secara umum, cara melakukan klasifikasi melalui pembelajaran mesin adalah dengan pertama membuat mesin belajar dari data latih. Data latih merupakan sekumpulan *instance* yang telah diberi label/kelas. Sebuah *instance* dibentuk oleh sekumpulan atribut. Contohnya, dalam klasifikasi makhluk hidup dapat dipilih atribut tempat hidup dan cara makan. Setiap *instance* makhluk hidup kemudian direpresentasikan dalam kedua atribut tersebut, misalnya $M1$(darat, membuat sendiri)

dan $M2$(laut, memakan makhluk hidup lain). Contoh dari label/kelas adalah binatang atau tumbuhan dan contoh data latih adalah $D : \{< M1, tumbuhan >, < M2, binatang >\}$. Sebuah acuan yang dipakai untuk membedakan satu kelas dengan kelas yang lain disebut sebagai konsep. Hal yang dilakukan oleh mesin selama pembelajaran adalah mencari fitur/ciri-ciri yang cocok kapan sebuah *instance* masuk ke dalam kelas tumbuhan dan kapan masuk ke dalam kelas binatang.

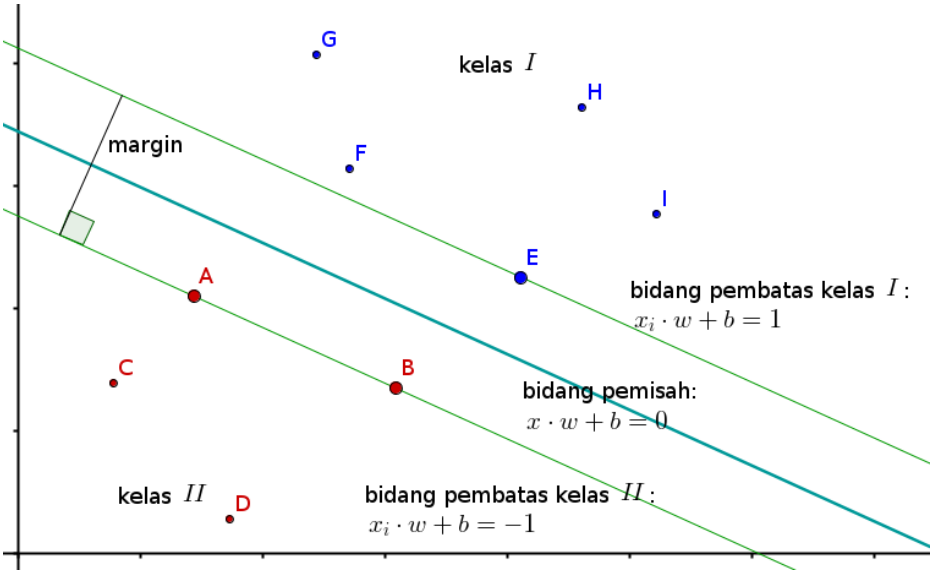
Setelah melakukan pembelajaran, mesin akan membuat sebuah hipotesis yang dapat dipakai untuk melakukan klasifikasi terhadap *instance* yang baru. Hipotesis tersebut hanya berupa pendekatan terhadap konsep dan akurasinya dapat meningkat/menurun bergantung pada berapa kali pembelajaran dilakukan dan sebagus apa data latih yang digunakan. Pada kenyataannya, tujuan utama dari pembelajaran mesin adalah mencari hipotesis terbaik. Pengukuran kinerja biasanya dilakukan dengan melakukan klasifikasi kembali data latih menggunakan hipotesis yang didapat. Selanjutnya, kinerja dapat diukur dengan menghitung berapa banyak *instance* yang terklasifikasi dengan benar dan berapa banyak yang tidak.

II.4 *Support Vector Machine*

SVM (*Support Vector Machine*) merupakan sebuah teknik pembelajaran mesin baru yang diperkenalkan oleh Boser, Guyon, dan Vapnik (1992). SVM dirancang untuk melakukan klasifikasi data ke dalam 2 kelas (*binary classification*), namun SVM dapat juga digunakan untuk kasus lebih dari 2 kelas dengan beberapa penyesuaian. SVM menggunakan algoritma pembelajaran yang disebut dengan *maximum margin training algorithm* (Boser dkk., 1992). Algoritma tersebut mencari *hyperplane* (bidang pembatas) dengan margin/jarak terbesar yang memisahkan kedua kelas. Ilustrasinya diberikan pada Gambar II.1.

SVM bekerja efektif pada data yang *linearly separable*. Secara geometri, hal tersebut berarti terdapat sebuah garis/bidang lurus yang dapat memisahkan kedua kelas. Bidang pembatas yang ingin dicari oleh SVM diberikan oleh persamaan vektor 2.1,

$$\vec{x} \cdot \vec{w} + b = 0 \tag{2.1}$$



Gambar II.1. Ilustrasi SVM (lingkaran yang lebih besar menandakan *support vector*)

dengan $\vec{w}$ adalah vektor yang tegak lurus dengan bidang pembatas dan b adalah jarak dari titik origin $O(0,0)$. Semua *instance* yang berada di atas bidang pemisah harus memenuhi $\vec{x}_i \cdot \vec{w} + b > 0$ sementara yang berada di bawah bidang pemisah harus memenuhi $\vec{x}_i \cdot \vec{w} + b < 0$. Sederhananya, fungsi yang menentukan kelas/label dari *instance* $\vec{x}_i$ diberikan oleh persamaan 2.2,

$$f(\vec{x}_i) = sign(\vec{x}_i \cdot \vec{w} + b) \tag{2.2}$$

dengan $sign(\vec{x}) = -1$ jika $x < 0$ dan $sign(\vec{x}) = +1$ jika $x > 0$.

Sebuah bidang pembatas harus menjaga konsistensi dari data latih. Maksudnya adalah setiap *instance* harus berada pada kelas yang sesuai dengan labelnya masing-masing. Hal tersebut dapat diperiksa menggunakan rumus 2.3,

$$y_i(\vec{x}_i \cdot \vec{w} + b) \geq 1 \tag{2.3}$$

dengan y_i menyatakan label dari *instance* $\vec{x}_i$. Selain itu, bidang pembatas juga harus memisahkan kedua kelas/label sejauh-jauhnya. Jarak antara bidang pembatas kedua

kelas disebut dengan margin. Besar margin didapatkan dengan rumus 2.4,

$$m = \frac{2}{|\vec{w}|^2} \tag{2.4}$$

dengan $|\vec{w}|$ menyatakan besar norma dari vektor $\vec{w}$.

Dengan penjelasan tersebut, permasalahan dapat dirangkum menjadi meminimalkan nilai dari $\frac{1}{2}|\vec{w}|^2$ dengan batasan $y_i(\vec{x_i} \cdot \vec{w} + b) \geq 1$. SVM menyelesaikan permasalahan tersebut dengan menentukan terlebih dahulu *instance* yang akan dilewati oleh bidang pembatas dari masing-masing kelas, yaitu semua *instance* yang memenuhi persamaan 2.5,

$$|\vec{x_i} \cdot \vec{w} + b| = 1. \tag{2.5}$$

Instance yang memenuhi persamaan tersebut disebut sebagai *support vector*.

Bagaimanapun, cara tersebut hanya akan berhasil menangani data yang *linearly separable*. Untuk data yang tidak *linearly separable*, SVM menggunakan fungsi pemetaan yang memetakan input $\vec{x_i}$ ke dimensi yang lebih tinggi (biasa dilambangkan dengan $\phi(\vec{x_i})$). Selain itu, dapat juga ditambahkan variabel eror ξ_i dan konstanta penalti C untuk menangani adanya *instance* dalam data latih yang berlabel salah. *Instance* yang seperti itu disebut dengan *noise*. Dengan demikian, permasalahan berubah menjadi meminimalkan nilai dari $\frac{1}{2}|\vec{w}|^2 + C\sum_i \xi_i$ dengan batasan $y_i(\phi(\vec{x_i}) \cdot \vec{w} + b) \geq 1 - \xi_i$.

Sayangnya, penggunaan fungsi pemetaan memakan biaya komputasi yang tinggi sehingga digunakanlah *kernel trick*, $K(\vec{x_i}, \vec{x_j}) = \phi(\vec{x_i})\phi(\vec{x_j})$, untuk mengurangi beban komputasi. Beberapa fungsi kernel yang umum digunakan diberikan pada Tabel II.4. Dengan memformulasikan permasalah menggunakan formula Lagrange, didapatkan fungsi untuk klasifikasi sebagai persamaan 2.6,

$$f(\vec{x}) = \sum_i \alpha_i y_i K(\vec{x_i}, \vec{x}) + b \tag{2.6}$$

dengan α disebut dengan $dual parameter$. *Support vector* dapat diidentifikasi melalui $\alpha_i > 0$. Hipotesis yang dihasilkan oleh SVM berupa kumpulan *support vector*

dengan nilai α-nya.

Tabel II.4. Fungsi kernel dalam SVM

Fungsi Kernel	**Formula ($K(x_i, x_j)$)**
Linear	$x_i^T x_j$
Polinomial	$\left(\gamma x_i^T x_j + r\right)^p, \quad \gamma > 0$
RBF	$\exp\left(-\gamma \left\lvert x_i - x_j \right\rvert^2\right), \quad \gamma > 0$
Sigmoid	$\tanh\left(\gamma x_i^T x_j + r\right)$

SVM sudah dicoba dalam banyak domain penelitian, seperti pengenalan tulisan tangan, klasifikasi teks, klasifikasi gambar, dan bioinformasi. Cristianini dan Shawe-Taylor (2000) menyebutkan bahwa SVM memiliki kinerja yang menjadi *state-of-the-art* dalam menyelesaikan persoalan dunia nyata. Selain itu, Vapnik (2000) juga memaparkan 3 kelebihan SVM dipandang dari sisi pembelajaran mesin, yaitu (1) bersifat universal karena menggunakan fungsi kernel yang dapat diubah sesuai kebutuhan, (2) memiliki batas atas eror yang kecil, dan (3) memiliki kinerja yang cepat.

II.5 Teknik Ekstraksi Istilah Dwibahasa

Pembuatan kamus dwibahasa secara otomatis dapat dilakukan dengan memanfaatkan ATR. ATR digunakan untuk melakukan ekstraksi istilah-istilah yang berpotensi menjadi kata yang penting (kata kunci) dan membentuk suatu daftar istilah yang koheren (logis dan konsisten). Daftar istilah tersebut nantinya akan menjadi entri kamus dwibahasa. Terdapat 2 jenis teknik yang dapat dipakai untuk melakukan ATR, yaitu (Ananiadou, 1994) (1) teknik linguistik dan (2) teknik nonlinguistik (statistik/probabilitas).

Saat ini, sudah banyak penelitian yang terkait dengan ekstraksi istilah dari korpus dengan berbagai macam teknik. Pada kenyataannya, kebanyakan teknik yang digunakan telah menggabungkan teknik linguistik dan nonlinguistik. Beberapa teknik/pendekatan yang berpotensi untuk diadopsi untuk bahasa Indonesia dipaparkan berikut ini.

II.5.1 Pendekatan Linguistik

Pendekatan linguistik memanfaatkan karakteristik bahasa, seperti Pos *tag*, pola, dan aturan bahasa (*grammar*). Salah satu penelitian mengenai ATR yang murni memakai pendekatan linguistik dilakukan oleh Ananiadou (1994). Tujuan penelitian tersebut adalah untuk mengekstrak istilah-istilah dalam domain farmasi. Idenya adalah dengan membentuk sebuah aturan pembentukan istilah/kata dalam level morfologi berupa awalan, akar kata, dan akhiran. Selain itu, Ananiadou (1994) juga mempertimbangkan banyaknya pemakaian awalan/kata/akhiran serapan dari bahasa Latin dan Yunani (*neoclassical element*) dalam pembuatan aturan.

Di samping itu, Tsuji dkk. (2002) memanfaatkan transliterasi untuk melakukan ekstraksi pasangan kata bahasa Jepang-Prancis. Idenya adalah dengan memanfaatkan kata bahasa Inggris yang diserap oleh kedua bahasa. Kata serapan biasanya memiliki ciri-ciri morfologi (huruf, bunyi huruf, dll) yang mirip dengan kata dari bahasa akarnya. Penelitian tersebut mengikuti penelitian sebelumnya yang menggunakan teknik yang sama namun ditujukan untuk pasangan kata bahasa Inggris-Jepang. Teknik tersebut menghasilkan tingkat presisi 80% untuk pasangan kata bahasa Jepang-Prancis. Teknik tersebut menjanjikan untuk diadopsi untuk bahasa Indonesia karena bahasa Indonesia juga menyerap banyak istilah dari bahasa asing (terutana Inggris), terutama dalam bidang pendidikan, pemerintahan, dan industri.

Bagaimanapun, terdapat batasan dalam pendekatan tersebut bahwa bahasa asal dan bahasa sasaran harus memiliki banyak kata yang seakar. Dengan kata lain, kinerja dari pendekatan ini bergantung pada karakteristik morfologi istilah dari masing-masing bahasa. Penelitian yang dilakukan oleh Fujii, Ishikawa, dan Lee (2004) memberikan hasil *trade-off* terhadap nilai presisi dan nilai *recall*. Sebagai contoh, presisi 50.0% memiliki *recall* 8.5% sedangkan *recall* 69.5% memiliki presisi hanya 1.2%. Hal tersebut dapat ditangani menggunakan sebuah kamus dwibahasa tambahan.

Secara umum, teknik tersebut dapat dibagi menjadi 2 buah proses utama, yaitu (1) pembangunan aturan transliterasi dan (2) ekstraksi pasangan kata transliterasi. Berikut ini dipaparkan langkah-langkah detail untuk setiap proses. Istilah "kata ka-

takana" digunakan untuk menyatakan kata dalam bahasa Jepang yang ditulis dalam huruf katakana (yang biasanya dipakai untuk menuliskan kata serapan).

II.5.1.1 Pembangunan Aturan Transliterasi

Salah satu hal krusial dalam teknik tersebut adalah harus tersedianya aturan transliterasi dari bahasa Jepang ke bahasa Prancis. Pembangunan aturan transliterasi dilakukan dengan cara

1. uraikan kata katakana dari sebuah daftar ke dalam kumpulan *mora unit*,
2. berdasarkan pasangan kata bahasa Prancisnya, buat aturan transliterasi untuk setiap *mora unit* secara manual, dan
3. ulangi langkah 1-2 untuk semua pasangan kata dalam daftar dan urutkan transliterasi berdasarkan frekuensi kemunculannya (yang paling sering yang paling pertama).

Daftar yang digunakan untuk pembangunan aturan transliterasi tidak hanya dari pasangan bahasa Jepang-Prancis, tetapi juga dari pasangan bahasa Inggris-Jepang. Daftar pasangan istilah bahasa Jepang-Prancis didapatkan dari *Concorde Japanese-French Dictionary* sedangkan pasangan istilah bahasa Inggris-Jepang didapatkan dari EDICT[12]. Tsuji dkk. (2002) menjelaskan bahwa penggunaan bahasa Inggris disebabkan pasangan istilah bahasa Inggris-Jepang lebih mudah ditemukan dan karakteristiknya tidak berbeda jauh dengan karakteristik bahasa Prancis.

II.5.1.2 Ekstraksi Pasangan Kata Transliterasi

Setelah didapatkan aturan transliterasi, korpora dwibahasa (Jepang-Prancis) digunakan untuk menentukan padanan kata bahasa Prancis diketahui sebuah kata dalam bahasa Jepang. Pencarian kata bahasa Prancis dihitung menggunakan ukuran *Dice* (yang melibatkan panjang karakter yang mirip) alih-alih komparasi karakter demi karakter. Hal tersebut disebabkan kata bahasa Prancis dalam korpus mungkin tidak muncul sebagai kandidat yang dibangkitkan dari aturan transliterasi. Langkah-langkahnya adalah sebagai berikut.

[12] `http://www.edrdg.org/jmdict/edict.html`

1. Ambil sebuah kata katakana, misalnya J, dan uraikan ke dalam kumpulan *mora unit*;
2. Dengan aturan transliterasi yang telah dibangun, bangkitkan semua kandidat transliterasi yang mungkin. Misalkan $T_i(J)$ menyatakan kandidat transliterasi ke-i;
3. Ambil kata bahasa Prancis, misalkan F, yang muncul bersamaan dengan J dalam korpus. Cari nilai *LCS*-nya dengan setiap $T_i(J)$ (misalkan dilambangkan dengan $LCS(F, T_i(J))$). *LCS* merupakan singkatan dari *Longest Common Subsequence* dan mengembalikan barisan karakter terpanjang yang muncul terurut (bukan berurutan) dalam kedua kata. Misalkan kedua kata tersebut adalah $X = x_1x_2 \ldots x_m$ dan $Y = y_1y_2 \ldots y_n$ dengan x_i, y_i melambangkan sebuah karakter. Misalkan juga $X_i = x_1x_2 \ldots x_i$ dan $Y_j = y_1y_2 \ldots y_j$. *LCS* (atau dapat disingkat dengan L) dapat dicari dengan rumus rekursi pada persamaan 2.7.

$$L(X_i, Y_j) = \begin{cases} \emptyset & \text{jika } i = 0 \text{ atau } j = 0 \\ L(X_{i-1}, Y_{j-1}) \cup x_i & \text{jika } x_i = y_j \\ lg\left(L(X_{i-1}, Y_j), L(X_i, Y_{j-1})\right) & \text{jika } x_i \neq y_j \end{cases} \tag{2.7}$$

   dengan fungsi lg mengembalikan barisan terpanjang;
4. Hitung nilai dari peluang p dengan persamaan 2.8:

$$P(J,F) = \max_i \left(Dice\left(len\left(LCS(T_i(J),F)\right), len\left(T_i(J)\right), len\left(F\right)\right)\right) \tag{2.8}$$

   dan ambil J dan F sebagai pasangan translasi jika nilainya melebihi *threshold*. Fungsi len menyatakan panjang karakter dari kata dan rumus *Dice* diberikan oleh persamaan 2.9.

$$Dice(k,m,n) = k \times \frac{2}{m+n}. \tag{2.9}$$

Sebagai ilustrasi (Tsuji dkk., 2002), misalkan aturan transliterasi diberikan oleh Tabel II.5 dan bahasa Jepang yang diekstrak adalah J : グラフ. Langkah-langkah selanjutnya adalah sebagai berikut.

1. J : グラフ $\Longrightarrow$ グ, ラ, フ.
2. Banyaknya kandidat transliterasi yang dibangkitkan ada $3 \times 6 \times 4 = 72$ buah

Tabel II.5. Contoh aturan transliterasi

グ	ラ	フ
g	ra	f
gue	la	phe
gu	l	ff
	lu	fe
	r	
	ler	

meliputi $graf, graphe, graff, grafe, \ldots, gulerff, gulerfe$.

3. Misalkan kata bahasa Prancis F : *graphe* muncul bersamaan dengan kata katakana tersebut.
4. Hitung nilai dari fungsi P dan ambil kandidat dengan nilai *Dice* tertinggi.

$$
\begin{aligned}
&P(\text{グラフ}, graphe) \\
&\quad = \max(Dice(len(LCS(graf, graphe)), len(graf), len(graphe)), \\
&\qquad Dice(len(LCS(graphe, graphe)), len(graphe), len(graphe)), \\
&\qquad Dice(len(LCS(graff, graphe)), len(graff), len(graphe)), \\
&\qquad \vdots \\
&\qquad Dice(len(LCS(gulerfe, graphe)), len(gulerfe), len(graphe))) \\
&\quad = \max(0.60, 1.00, 0.55, \ldots, 0.31) \\
&\quad = 1.00
\end{aligned}
$$

sehingga グラフ dan *graphe* kemungkinan besar merupakan sebuah pasangan istilah yang ekivalen.

II.5.2 Pendekatan Statistik

Pendekatan statistik menggunakan informasi *n-gram* dan perhitungan statistik seperti tingkat persebaran kata dalam dokumen. Salah satu contohnya adalah penelitian yang dilakukan oleh Rapp (1995). Idenya adalah jika kata A dan B muncul bersama lebih sering dari yang diharapkan dalam bahasa asal, translasi keduanya dalam bahasa sasaran juga akan muncul bersama-sama lebih sering dari yang diharapkan.

Teknik tersebut mengasumsikan bahwa terdapat korelasi kemunculan kata dalam teks yang berbeda bahasa. Rapp (1995) merepresentasikan frekuensi kemunculan bersama sebagai sebuah matriks berukuran $n \times n$ (*n-gram*). Dua buah matriks akan menjadi pasangan translasi satu dengan yang lain jika kedua matriks tersebut memiliki jarak *city-block* yang terdekat.

Berbeda dengan Rapp (1995), Yu dan Tsujii (2009) justru menggunakan ukuran *context heterogeneity similarity* yang diadopsi dari penelitian yang dilakukan oleh Fung (1995). Fung (1995) mengajukan bahwa *context heterogeneity similarity* adalah fitur/ciri-ciri yang lebih menonjol (*salient*) dibandingkan frekuensi kemunculan bersama.

Konsepnya adalah sebuah kata/istilah dalam sebuah domain hanya memiliki kata-kata tertentu yang dapat langsung mendahului atau mengikutinya. Semakin umum sebuah kata/istilah (bukan istilah penting dalam domain), semakin banyak kata yang dapat langsung mendahului atau mengikutinya. Dengan demikian, nilai *context heterogeneity similarity* dari kata tersebut semakin besar. Sebaliknya, istilah dalam domain hanya memiliki sedikit kata yang dapat langsung mendahului atau mengikutinya sehingga nilai *context heterogeneity similarity* dari kata tersebut kecil. *Context heterogeneity similarity* dari sebuah kata W dinyatakan sebagai sebuah vektor 2 dimensi $\vec{h}$ sebagaimana yang ditunjukkan pada persamaan 2.10:

$$\vec{h} = (h_{kiri}, h_{kanan}) \tag{2.10}$$

dengan

$$h_{kiri}(W) = \frac{\text{banyak kata berbeda yang langsung mendahului } W}{\text{banyak kemunculan } W} \quad \text{dan}$$
$$h_{kanan}(W) = \frac{\text{banyak kata berbeda yang langsung mengikuti } W}{\text{banyak kemunculan } W}.$$

Yu dan Tsujii (2009) kemudian menyempurnakan fitur tersebut dengan menambah fitur *dependency heterogeneity similarity*. Fitur tersebut diajukan untuk menangani masalah terdapat kata yang muncul dalam konteks yang serupa, namun bukan pasangan translasi. *Dependency heterogeneity similarity* dari sebuah kata W dinya-

takan sebagai sebuah vektor 4 dimensi $\vec{d}$ sebagaimana yang ditunjukkan pada persamaan 2.11:

$$\vec{d} = (d_{NMODHead}, d_{SUBHead}, d_{OBJHead}, d_{NMODMod}) \tag{2.11}$$

dengan

$$d_{NMODHead} = \frac{\text{banyak } \textit{head} \text{ berbeda dari } W \text{ dengan label NMOD}}{\text{banyak } \textit{head} \text{ dari } W \text{ dengan label NMOD}},$$
$$d_{SUBHead} = \frac{\text{banyak } \textit{head} \text{ berbeda dari } W \text{ dengan label SUB}}{\text{banyak } \textit{head} \text{ dari } W \text{ dengan label SUB}},$$
$$d_{OBJHead} = \frac{\text{banyak } \textit{head} \text{ berbeda dari } W \text{ dengan label OBJ}}{\text{banyak } \textit{head} \text{ dari } W \text{ dengan label OBJ}}, \text{ dan}$$
$$d_{NMODmod} = \frac{\text{banyak } \textit{modifier} \text{ berbeda dari } W \text{ dengan label NMOD}}{\text{banyak } \textit{modifier} \text{ dari } W \text{ dengan label NMOD}}.$$

Penentuan pasangan translasi dilakukan dengan menghitung nilai *context heterogeneity similarity* dan *dependency heterogeneity similarity* untuk setiap kandidat istilah dari masing-masing bahasa. Penentuan pasangan translasi atau bukan dilakukan dengan mengukur jarak *euclidean*. Pasangan istilah yang diambil sebagai pasangan translasi adalah pasangan istilah dengan jarak *euclidean* terkecil.

Aker dkk. (2013) menggunakan pendekatan yang berbeda dari yang lain, yaitu dengan memandang ATR sebagai permasalahan klasifikasi. Hasil yang didapatkan adalah hingga 83% pasangan istilah yang dibangkitkan merupakan translasi yang tepat. Pendekatan ini mengasumsikan bahwa terdapat cara untuk mengekstrak *monolingual term* baik untuk bahasa asal maupun untuk bahasa sasaran. Dibandingkan pendekatan linguistik yang dibahas pada Subbab II.5.1, teknik yang dipakai dalam pendekatan ini lebih sederhana. Ringkasnya, cara yang digunakan adalah

1. memasangkan setiap istilah yang diekstrak dari korpus bahasa asal dengan setiap istilah yang diekstrak dari korpus bahasa sasaran,
2. mengekstrak fitur dari setiap pasangan istilah, dan
3. berdasarkan fitur yang diekstrak, melakukan klasifikasi apakah pasangan istilah tersebut ekuivalen atau tidak (*binary classification*).

Fitur yang dipakai dapat dibagi menjadi 3 kategori, yaitu (1) fitur berbasis kamus,

(2) fitur berbasis kognat, dan (3) gabungan keduanya. Untuk mengekstrak fitur berbasis kamus, dibutuhkan sebuah kamus dwibahasa. Fitur berbasis kamus memiliki arah, yaitu bergantung pada pemilihan mana yang menjadi bahasa asal dan mana yang menjadi bahasa sasaran. Oleh karena itu, total fitur berbasis kamus ada $2 \times 6 + 1 = 13$ buah dengan satu fitur adalah rata-rata dari 2 kemungkinan pemilihan pasangan bahasa asal-sasaran sebagaimana yang dijelaskan berikut ini.

1. *isFirstWordTranslated*: fitur *boolean* yang menandakan apakah kata pertama dari istilah bahasa asal merupakan translasi kata pertama dari istilah bahasa sasaran. Contohnya, pasangan istilah (automatic term recognition, pengenalan istilah otomatis) bernilai $false$ sedangkan pasangan istilah (rule of thumb, aturan ibu jari) bernilai $true$.
2. *isLastWordTranslated*: fitur *boolean* yang menandakan apakah kata terakhir dari istilah bahasa asal merupakan translasi kata terakhir dari istilah bahasa sasaran. Contohnya, pasangan istilah (automatic term recognition, pengenalan istilah otomatis) bernilai $false$ sedangkan pasangan istilah (rule of thumb, aturan ibu jari) bernilai $true$.
3. *percentageOfTranslatedWords*: persentasi banyak kata dari istilah bahasa asal yang memiliki translasi dalam istilah bahasa sasaran. Contohnya, pasangan istilah (automatic term recognition, pengenalan istilah) bernilai $\frac{2}{3} \times 100\% = 66,57\%$.
4. *percentageOfNotTranslatedWords*: persentasi banyak kata dari istilah bahasa asal yang tidak memiliki translasi dalam istilah bahasa sasaran. Contohnya, pasangan istilah (automatic term recognition, pengenalan istilah) bernilai $\frac{1}{3} \times 100\% = 33,33\%$.
5. *longestTranslatedUnitInPercentage*: rasio (dalam persen) banyak kata berurutan yang terpanjang dari istilah bahasa asal yang memiliki translasi dalam istilah bahasa sasaran terhadap banyak kata seluruhnya dari istilah bahasa asal. Contohnya, pasangan istilah (responsive web user interface development, pengembangan antarmuka yang responsif) bernilai $\frac{2}{5} \times 100\% = 40\%$.
6. *longestNotTranslatedUnitInPercentage*: rasio (dalam persen) banyak kata berurutan yang terpanjang dari istilah bahasa asal yang tidak memiliki translasi

dalam istilah bahasa sasaran terhadap banyak kata seluruhnya dari istilah bahasa asal. Contohnya, pasangan istilah (responsive web user interface development, antarmuka yang responsif) bernilai $\frac{2}{5} \times 100\% = 40\%$.

7. *averagePercentageOfTranslatedWords*, yaitu rata-rata dari *percentageOfTranslatedWords* untuk 2 kemungkinan pemilihan pasangan bahasa asal-sasaran. Contohnya, Contohnya, pasangan istilah (responsive web user interface development, pengembangan antarmuka yang responsif) memiliki nilai *percentageOfTranslatedWords* $\frac{2}{5} \times 100\% = 40\%$ sementara pasangan istilah (pengembangan antarmuka yang responsif, responsive web user interface development) memiliki nilai $\frac{2}{4} \times 100\% = 50\%$. Jadi, nilai dari *averagePercentageOfTranslatedWords* adalah $(40\% + 50\%)/2 = 45\%$.

Fitur berbasis kognat digunakan untuk menangani istilah yang tidak dapat ditangani oleh kamus, seperti *named entity* dan istilah khusus. Aker dkk. (2013) menyebutkan, "Metode kognat mengasumsikan bahwa bahasa asal dan bahasa sasaran yang dibandingkan menggunakan jenis karakter yang sama". Oleh karena itu, dibuat sebuah pemetaan karakter dari bahasa asal ke bahasa sasaran dan sebaliknya jika kedua bahasa menggunakan karakter yang berbeda. Terdapat 5 buah fitur kognat, sehingga totalnya terdapat $5 + 2 \times 5 = 15$ berupa 5 fitur tanpa pemetaan karakter, 5 fitur dengan karakter bahasa asal, dan 5 fitur dengan karakter bahasa sasaran. Berikut ini adalah detail fitur berbasis kognat yang digunakan.

1. *Longest Common Subsequence Ratio* (LCSR):

$$LCSR(X,Y) = \frac{len(LCS(X,Y))}{\max(len(X), len(Y))} \tag{2.12}$$

dengan $LCS(X,Y)$ mengembalikan barisan karakter terpanjang yang muncul terurut (bukan berurutan) dalam kedua kata. Contohnya,

$$LCS(efisien, efficient) = efiien$$

maka

$$\begin{aligned} LCSR(efisien, efficient) &= \frac{len(efiien)}{\max(len(efisien), len(efficient))} \\ &= \frac{6}{\max(7,9)} \\ &= \frac{6}{9} \\ &= 0,67. \end{aligned}$$

2. *Longest Common Substring Ratio* (LCSTR):

$$LCSTR(X,Y) = \frac{len(LCST(X,Y))}{\max(len(X), len(Y))} \tag{2.13}$$

dengan $LCST(X,Y)$ menyatakan *substring* (barisan karakter berurutan) terpanjang yang ada baik dalam X maupun Y. Contohnya,

$$LCST(efisien, efficient) = ien$$

maka

$$\begin{aligned} LCSTR(efisien, efficient) &= \frac{len(ien)}{\max(len(efisien), len(efficient))} \\ &= \frac{3}{\max(7,9)} \\ &= \frac{3}{9} \\ &= 0,33. \end{aligned}$$

3. *Dice Similarity*:

$$dice = 2 \times \frac{len(LCST(X,Y))}{len(X) + len(Y)}. \tag{2.14}$$

4. *Needleman Wunsch Distance* (NWD):

$$NWD = \frac{len(LCST(X,Y))}{\min(len(X), len(Y))}. \tag{2.15}$$

5. *Levenshtein Distance* (LD):

$$LD_{normalized}(X,Y) = 1 - \frac{LD(X,Y)}{\max(len(X), len(Y))}. \tag{2.16}$$

Sebenarnya, *levenshtein distance* menyatakan banyak operasi-satu-karakter minimum yang harus dilakukan untuk mengubah satu string menjadi *string* yang lain. Operasi yang terdefinisi antara lain *insertion* (menambah 1 karakter), *deletion* (menghapus 1 karakter), dan *substitution* (mengubah 1 karakter). Untuk penyeragaman dengan fitur kognat yang lain, Aker dkk. (2013) menormalkannya menjadi bentuk tersebut. Contohnya,

$$LD(efisien, efficient) = 3$$

meliputi 2 operasi *insertion* (huruf 'f' dan 't') dan 1 operasi *substitution* (huruf 's' menjadi huruf 'c'). Jadi,

$$\begin{aligned} LD_{normalized}(efisien, efficient) &= 1 - \frac{LD(efisien, efficient)}{\max(len(efisien), len(efficient))} \\ &= 1 - \frac{3}{\max(7,9)} \\ &= 0,67. \end{aligned}$$

Fitur berbasis gabungan kamus dan kognat juga dipertimbangkan dalam pembelajaran. Untuk mengekstrak fitur tersebut, didefinisikan bahwa sebuah kata dari bahasa asal memiliki translasi dalam bahasa sasaran jika terdapat pasangan kata tersebut di dalam kamus. Didefinisikan juga bahwa sebuah kata dari bahasa asal memiliki transliterasi dalam bahasa sasaran jika salah satu fitur kognatnya bernilai lebih dari $0,7$. Fitur gabungan juga memiliki arah, sehingga totalnya ada $2 \times 5 = 10$ fitur gabungan. Berikut ini adalah daftar fitur gabungan yang dipakai.

1. *isFirstWordCovered*: fitur bernilai *boolean* yang menandakan apakah kata pertama dari istilah bahasa asal memiliki translasi atau transliterasi dalam istilah bahasa sasaran.
2. *isLastWordCovered*: fitur bernilai *boolean* yang menandakan apakah kata terakhir dari istilah bahasa asal memiliki translasi atau transliterasi dalam istilah bahasa sasaran.
3. *percentageOfCoverage*: persentasi dari banyak kata dari istilah bahasa asal yang memiliki translasi atau transliterasi dalam istilah bahasa sasaran.
4. *percentageOfNonCoverage*: persentasi dari banyak kata dari istilah bahasa

asal yang tidak memiliki translasi atau transliterasi dalam istilah bahasa sasaran.

5. *difBetweenCoverageAndNonCoverage*: selisih dari 2 fitur terakhir (*percentageOfCoverage* dan *percentageOfNonCoverage*).

II.6 Penelitian Terkait

Pembangunan kamus dwibahasa Indonesia-Jepang pernah dilakukan sebelumnya oleh Limanthie (2014). Teknik yang digunakannya cukup intuitif, yaitu dengan melihat seberapa sering sepasang istilah bahasa Indonesia-Jepang muncul bersamaan di dalam korpus. Pasangan istilah dengan frekuensi kemunculan bersama yang tinggi tentunya cenderung merupakan pasangan translasi. Terdapat 6 variabel yang diperhatikan dalam eksperimen yang dilakukannya, yaitu (1) ada-tidaknya *stop word*, (2) lingkup korpus: per dokumen atau seluruh dokomen, (3) jarak antarkata, (4) jenis kata: tidak dibatasi atau kata benda saja, (5) metode perhitungan frekuensi yang dipakai: *single co-occurrence* atau *multi co-occurrence*, dan (6) menambahkan judul artikel ke dalam kamus awal atau tidak.

Eksperimen dilakukan dengan korpora *comparable* yang dikumpulkan dari Wikipedia dengan domain *computer science*. Terdapat 152 pasangan artikel bahasa Indonesia-Jepang yang digunakan dalam pengujian (Limanthie, 2014). Pemrosesan seluruh dokumen memiliki sisi positif dapat lebih banyak pasangan istilah yang terekstrak, namun konteks dari 152 artikel tersebut tercampur. Di lain pihak, pemrosesan per dokumen memiliki konteks yang fokus sehingga dapat memberikan hasil yang lebih tepat, namun pasangan istilah yang terekstrak kurang beragam.

Selain itu, Limanthie (2014) juga menggunakan kamus awal untuk membantu mencari kandidat pasangan istilah. Entri kamus berupa kata-kata umum dan dipakai sebagai acuan (*anchor*) untuk membangkitkan kandidat pasangan istilah. Contohnya, diketahui dari kamus awal bahwa kata "bahasa" memiliki padanan kata "言語" dalam bahasa Jepang. Dari korpus, didapatkan kata "pemrograman" sering muncul berdekatan dengan kata "bahasa" sedangkan kata "プログラミング" sering muncul berdekatan dengan kata "言語". Secara intuitif, dapat disimpulkan bahwa

pasangan kata (pemrograman,プログラミング) mungkin sebuah pasangan translasi dan dapat diambil sebagai sebuah kandidat. Kamus awal tentunya masih memiliki sedikit entri dan dapat diperbanyak dengan menambahkan judul artikel ke dalam kamus. Hal tersebut mungkin berhasil karena judul artikel Wikipedia biasanya adalah translasi langsung satu dengan yang lain.

Terdapat 2 pendekatan yang dilakukan Limanthie (2014) untuk menghitung frekuensi kemunculan, yaitu (1) *single co-occurrence* dan (2) *multi co-occurrence*. *Single co-occurrence* hanya menggunakan satu kata acuan sedangkan *multi co-occurrence* menggunakan 2 atau lebih kata acuan. Limanthie (2014) kemudian membuat 7 buah eksperimen yang ditujukan untuk mencari kondisi yang paling kondusif untuk melakukan ekstraksi istilah. Ketujuh kondisi tersebut beserta hasilnya diberikan pada Tabel II.6.

Tabel II.6. Hasil eksperimen yang dilakukan Limanthie (2014)

No.	**Perubahan**	**Hasil**
1	Akurasi awal.	Tidak ada.
2	Menghilangkan *stop word*.	Lebih baik *stop word* dihilangkan.
3	Mengubah pembentukan daftar pasangan kata menjadi per dokumen.	Lebih baik lingkup korpus per dokumen.
4	Memproses kata benda saja.	Lebih baik hanya kata benda yang diproses untuk jarak 3 kata.
5	Mempertimbangkan *multi co-occurrence*.	Lebih baik *single co-occurrence*.
6	Menambah judul artikel pada kamus awal.	Tidak ada perbedaan.
7	Melakukan iterasi.	Lebih baik dengan iterasi.

Selain teknik tersebut, pembuatan kamus dwibahasa juga dapat dilakukan dengan bantuan pembelajaran mesin. Pendekatan tersebut dilakukan oleh Aker dkk. (2013). Idenya adalah dengan memandang translasi sebagai masalah klasifikasi: apakah sebuah pasangan istilah ekuivalen atau tidak. Detail mengenai teknik yang digunakan Aker dkk. (2013) telah dijelaskan pada Subbab II.5.2. Klasifikasi dilakukan

menggunakan SVM *binary classifier* dengan data latih diambil dari EUROVOC[13]. EUROVOC merupakan sebuah ensiklopedi (*thesaurus*) multibahasa dan multidisiplin yang mencakup aktivitas dari *European Union*, terutama *European Parliament*. EUROVOC kini melingkupi 23 bahasa berbeda negara-negara di benua Eropa.

Aker dkk. (2013) juga menggunakan sebuah kamus untuk mengekstrak fitur berbasis kamus. Pembuatan kamus dilakukan dengan GIZA++[14]. Kamus yang dihasilkan merupakan kumpulan entri dengan format $< s, t_i, p_i >$ dengan s menyatakan kata dari bahasa asal, t_i adalah translasi ke-i dari s dalam bahasa sasaran, dan p_i adalah peluang s ditranslasikan dengan t_i. Kemudian, kamus disaring dengan menghapus entri yang memiliki nilai peluang $p_i < 0.05$ dan entri yang mengandung *stop word*. Contoh keluaran dari GIZA++ diberikan pada Tabel II.7.

Tabel II.7. Contoh entri kamus latih

Bahasa Indonesia	**Bahasa Jepang**	**Peluang**
bahasa	言語	0,68
bahasa	語	0,45
kendaraan	車	0,66
komputer	コンピュータ	0,99
mobil	車	0,50

Pendekatan yang dilakukan Aker dkk. (2013) tidak bergantung pada jenis korpora yang dipakai. Selain itu, jenis kata yang diproses tidak hanya kata benda, tetapi juga bisa kata kerja atau kata sifat. Meskipun demikian, pendekatan tersebut memerlukan *monolingual term extractor* untuk masing-masing bahasa. Oleh karena itu, kinerjanya sangat bergantung pada kualitas daftar istilah yang berhasil diekstrak dari masing-masing bahasa.

Hasil akhir dari penelitian yang dilakukan Limanthie (2014) masih memiliki akurasi kurang dari 5% dengan banyaknya pasangan istilah translasi yang benar yang berhasil diekstrak mencapai 80 pasangan dari semua eksperimen. Limanthie (2014) menyebutkan hal tersebut disebabkan korelasi antarartikel dalam Wikipedia yang berbeda versi bahasa sangat kecil karena beda artikel ditulis oleh beda orang. Penyebab lainnya adalah karena banyak kata memiliki kandidat translasi yang lebih

[13] `http://eurovoc.europa.eu/drupal/?q=download/list_pt&cl=en`
[14] `https://code.google.com/p/giza-pp/`

dari satu, padahal kebanyakan hanya satu yang benar. Hal tersebut tentunya mengurangi akurasi kamus yang dibangun. Di lain pihak, penelitian yang dilakukan Aker dkk. (2013) mendapatkan hasil yang bagus. Sebanyak 80% dari pasangan istilah yang berhasil diekstrak merupakan pasangan translasi yang ekivalen.

BAB III

PEMBANGUNAN KAMUS INDONESIA-JEPANG

III.1 Analisis Masalah

Berdasarkan masalah yang telah dirumuskan pada Subbab I.2, tugas akhir ini ditujukan untuk dapat menangani kondisi korpus yang sedikit dan belum ditemukannya teknik yang cukup bagus untuk mengolah bahasa Indonesia. Berikut ini dipaparkan analisis yang lebih detail terhadap kedua permasalahan tersebut.

III.1.1 Analisis Korpus

Limanthie (2014) menggunakan korpus *comparable* yang diambil dari Wikipedia dalam penelitiannya. Sayangnya, hasil yang didapatkan belum memuaskan. Limanthie (2014) menganalisis bahwa hal tersebut disebabkan artikel Wikipedia yang berbeda versi bahasa ditulis oleh orang yang berbeda juga, sehingga peluang isi keduanya berbeda sangat besar. Mengingat bahwa teknik yang digunakannya berdasarkan frekuensi kemunculan bersama, yang menggunakan *n-gram*, dapat disimpulkan bahwa korpus dari Wikipedia tidak dapat diolah per sekian kata-sekian kata. Dengan kata lain, penggunaan *n-gram* saja tidak tepat.

Melihat kenyataan tersebut, korpus *comparable* tidak bagus digunakan sendirian untuk menyelesaikan masalah ATR. Banyak penelitian yang menggunakan kamus dwibahasa yang telah didefinisikan sebelumnya sebagai bantuan. Permasalahannya kemudian adalah tidak adanya kamus awal yang cukup besar untuk pasangan bahasa Indonesia-Jepang. Hal tersebut juga dihadapi Limanthie (2014) dalam tugas akhirnya. Solusi dari Limanthie (2014) adalah dengan menambahkan judul artikel ke dalam kamus awal, namun tidak ada perubahan kinerja yang signifikan.

Di lain pihak, Aker dkk. (2013) menyelesaikan masalah tersebut dengan cara yang berbeda. Entri kamus dibangkitkan dengan mengekstrak pasangan istilah dari korpus *parallel*. Ekstraksi dilakukan menggunakan perangkat lunak GIZA++[15] sehingga tidak perlu ditangani sendiri lagi. Selain itu, kamus awal tidak perlu memiliki

[15] `https://code.google.com/p/giza-pp/`

domain yang sama dengan korpus yang akan diuji. Dengan demikian, prasyarat korpus *parallel* semakin longgar. Beberapa korpus *parallel* dwibahasa Indonesia-Jepang dapat ditemukan pada OPUS[16].

III.1.2 Analisis Teknik Ekstraksi

Sebagaimana yang telah dijelaskan pada Subbab III.1.1, korpus yang diambil dari Wikipedia tidak cocok ditangani hanya dengan *n-gram*. Hal tersebut disebabkan korpora *comparable* tidak memiliki pola yang menentu, sehingga penggunaan *n-gram* akan melewatkan istilah-istilah ekivalen di luar jangkauan n. Contohnya, pencarian padanan kata "sistem" dengan $n = 5$ hanya akan melingkupi 5 kata sebelumnya dan 5 kata setelahnya. Andaikan padanan kata dalam bahasa Jepangnya, yaitu "システム", muncul pada kata ke-6, pasangan istilah tersebut tidak akan terdeteksi. Dengan kata lain, *n-gram* hanya melihat secara lokal. Oleh karena itu, dibutuhkan teknik yang dapat melihat korpus secara menyeluruh/global.

Teknik yang dipakai oleh Tsuji dkk. (2002) dapat dipakai untuk menyelesaikan masalah tersebut sebagaimana yang telah dijelaskan pada Subbab II.5.1. Dengan teknik ini, hanya dibutuhkan sebuah korpus dwibahasa Indonesia-Jepang (tidak peduli apakah *parallel* atau *comparable*) dan sebuah aturan transliterasi. Aturan transliterasi dibangkitkan menggunakan kamus Indonesia-Jepang yang sudah ada. Kamus yang digunakan tidak harus berdomain spesifik, tetapi cukup kamus umum saja sehingga mudah ditemukan. Bagaimanapun, teknik ini hanya akan mengekstrak istilah-istilah dari bahasa Indonesia dan Jepang yang memiliki akar bahasa yang sama. Dalam domain yang spesifik, terdapat banyak istilah serapan dalam kedua bahasa yang diambil dari bahasa Inggris sehingga teknik ini menjanjikan untuk dipakai.

Teknik yang dipakai oleh Aker dkk. (2013) juga dapat dipakai untuk menyelesaikan masalah ini. Penjelasan tekniknya telah dipaparkan pada Subbab II.5.2 dan beberapa detail teknisnya diberikan pada Subbab II.6. Dengan teknik ini, juga hanya dibutuhkan sebarang korpus dwibahasa Indonesia-Jepang yang tidak harus *parallel*

[16] `http://opus.lingfil.uu.se/`

atau *comparable*. Keistimewaan teknik ini adalah digabungkannya aspek linguistik dan aspek statistik dalam penentuan pasangan istilah yang menjadi translasi. Bagaimanapun, teknik ini mengharuskan terdapat *monolingual term extractor* baik untuk bahasa Indonesia maupun untuk bahasa Jepang.

Monolingual term extractor digunakan untuk mengekstrak istilah-istilah dari masing-masing bahasa. Kemudian, setiap istilah dari bahasa Indonesia dipasangkan dengan setiap istilah bahasa Jepang. Dengan demikian, masalah berubah menjadi menentukan apakah sebuah pasangan istilah merupakan translasi satu dengan yang lain atau bukan. Teknik ini juga menggunakan kamus awal untuk mengekstrak fitur berbasis kamus. Kamus awal dibangun dari sebarang korpus *parallel* menggunakan GIZA++. Untuk hal tersebut, OPUS telah menyediakan korpus *parallel* yang dibutuhkan untuk pasangan bahasa Indonesia-Jepang.

Kelebihan lainnya dari pendekatan yang digunakan Aker dkk. (2013) adalah digunakannya pembelajaran mesin. Pembelajaran mesin cocok untuk mencari sebuah pola yang tidak dapat ditangani secara manual. Oleh karena itu, pendekatan tersebut cocok untuk kondisi bahasa Indonesia yang belum banyak tereksploitasi. Di sisi lain, pembelajaran mesin mengharuskan terdapat data latih yang dapat dipakai untuk melatih mesin pertama kali. Dalam tugas akhir ini, data latih yang dimaksud adalah kumpulan pasangan istilah dwibahasa Indonesia-Jepang. Hal yang diperlukan dalam membangun data latih adalah harus terdapat cukup banyak entri pasangan istilah dan semua entri cukup mewakili semua kondisi kemungkinan translasi dari bahasa Indonesia ke bahasa Jepang (atau sebaliknya).

Selain itu, teknik yang digunakan oleh Aker dkk. (2013) dapat diintegrasikan dengan teknik yang dipakai oleh Yu dan Tsujii (2009). Fitur *context heterogeneity similarity* dan *dependency heterogeneity similarity* dapat digabungkan ke dalam fitur-fitur yang dipakai dalam pembelajaran mesin. Teknik tersebut sebetulnya sudah dijadikan acuan dalam tugas akhir Limanthie (2014), tetapi dalam implementasinya Limanthie (2014) tidak menggunakannya. Oleh karena itu, kedua fitur tersebut mungkin dapat meningkatkan presisi kamus yang dibangkitkan.

Teknik yang dipakai Limanthie (2014) dalam tugas akhirnya adalah teknik yang

diadopsi dari penelitian Rapp (1995). Teknik tersebut dalam tugas akhir ini tidak dipakai karena 2 alasan berikut:

1. teknik tersebut mengasumsikan terdapat kemiripan pola kemunculan kata antara dokumen bahasa asal dengan bahasa sasaran (Rapp, 1995). Di sisi lain, Limanthie (2014) telah menunjukkan bahwa korpus yang dipakai dari Wikipedia memiliki tingkat korelasi yang rendah atau dengan kata lain memiliki pola yang cukup berbeda agar teknik tersebut dapat bekerja dengan efektif; dan
2. penggunaan *context heterogeneity similarity* dapat menggantikan frekuensi kemunculan bersama dan lebih efektif sebagaimana yang dikemukakan oleh Fung (1995).

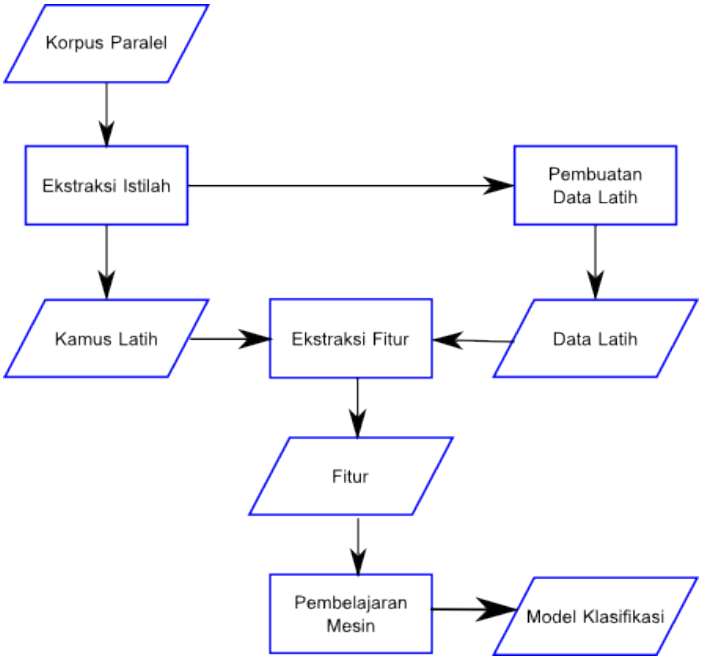
III.2 Analisis Solusi

III.2.1 Teknik dan Desain Sistem

Berdasarkan analisis masalah pada Subbab III.1, teknik yang paling baik untuk digunakan adalah teknik yang digunakan Aker dkk. (2013). Tentunya, perlu beberapa perubahan untuk menyesuaikan dengan ciri-ciri bahasa Indonesia dan Jepang. Sistem pembangunan kamus dapat dibagi ke dalam 2 proses utama, yaitu (1) pelatihan mesin dan (2) ekstraksi korpus.

Pelatihan mesin ditujukan untuk menentukan hipotesis yang dapat dipakai untuk membedakan apakah sebuah pasangan istilah merupakan pasangan translasi atau bukan. Sebelum proses ini dilakukan, dibuat terlebih dahulu sebuah kamus latih dan sebuah data latih. Kamus latih dibangkitkan dari sebuah korpus/istilah yang telah *parallel*. Kamus latih digunakan sebagai alat bantu untuk mengekstrak fitur berbasis kamus. Data latih digunakan sebagai acuan penentuan fungsi hipotesis. Keluaran dari proses ini adalah sebuah model klasifikasi. Alur pelatihan mesin beserta dengan praprosesnya diberikan pada Gambar III.1.

Pada tahap ekstraksi korpus, dilakukan ekstraksi istilah menggunakan GIZA++ dari korpus dwibahasa. Secara ringkas, GIZA++ memasangkan setiap *n* kata (*n-gram*) dari bahasa asal dengan setiap *n* kata dari bahasa sasaran. Selain itu, dihitung pula



Gambar III.1. Desain sistem pembelajaran

frekuensi kemunculannya pada korpus yang dapat dianggap sebagai peluang kemunculan kata tersebut. Alur ekstraksi istilah dari korpus diberikan pada Gambar III.2.

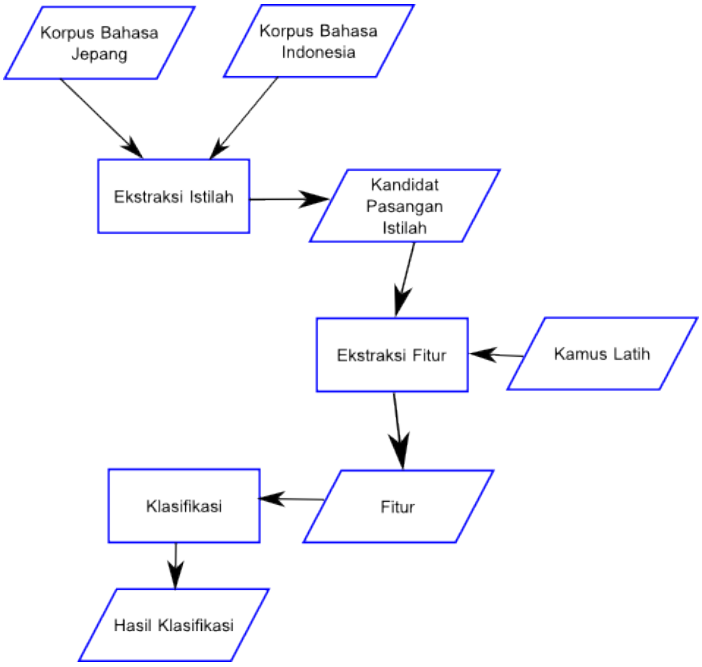
III.2.2 Pembuatan Data Latih

Data latih diperlukan sebagai bahan pembelajaran. Data latih berupa sebuah daftar pasangan padanan kata/frasa bahasa Indonesia-Jepang yang sudah terdefinisi beserta labelnya (benar/salah). Domain data latih tidak harus sama dengan domain yang dipakai ketika pengujian. Data latih untuk pasangan kata/frasa bahasa Indonesia-Jepang tidak tersedia banyak (atau bahkan tidak ada sama sekali) sehingga perlu penanganan khusus untuk membuatnya. Contoh data latih diberikan pada Tabel III.1.

Tabel III.1. Contoh entri data latih

Bahasa Indonesia	**Bahasa Jepang**	**Peluang**
bahasa	言語	benar
komputer	プログラミング	salah
mobil	車	benar

Pembangunan data latih dilakukan dalam 2 proses, yaitu (1) ekstraksi istilah dari



Gambar III.2. Desain sistem ekstraksi istilah dari korpus

korpus *parallel* dan (2) pelabelan benar/salah secara manual dari hasil yang didapatkan. Korpus *parallel* didapatkan dari OPUS sementara ekstraksi istilah dilakukan menggunakan GIZA++. Contoh keluaran GIZA++ diberikan pada Tabel II.7. Pelabelan benar/salah dilakukan secara manual dengan melihat ke kamus umum bahasa Indonesia-Jepang berdasarkan nilai peluang yang didapatkan dari GIZA++.

III.2.3 Pembuatan Kamus Latih

Kamus latih diperlukan untuk mengekstrak fitur berbasis kamus. Aker dkk. (2013) menggunakan GIZA++ untuk membuat kamus latih dengan data diambil dari DGT-TM (*DGT-Translation Memory*)[17]. DGT-TM merupakan daftar istilah dalam 22 bahasa berbeda di Eropa yang disusun secara paralel.

Entri dari kamus latih berupa sebuah pasangan kata bahasa Indonesia I dengan kata bahasa Jepang J beserta nilai peluang p dengan p adalah peluang I ditranslasikan menjadi J. Contohnya diberikan pada tabel II.7. Pembuatan kamus latih untuk tugas akhir ini mengikuti cara yang dipakai oleh Aker dkk. (2013), yaitu menggunakan GIZA++ sedangkan Korpus *parallel* diambil dari OPUS.

[17] `https://ec.europa.eu/jrc/en/language-technologies/dgt-translation-memory`

III.2.4 Penyesuaian

Fitur kognat yang digunakan oleh Aker dkk. (2013) terbagi menjadi 2, yaitu (1) yang memiliki karakter sama dan (2) yang memiliki karakter berbeda. Karena karakter bahasa Indonesia dan bahasa Jepang berbeda, fitur kognat yang memperhitungkan karakter yang sama sudah tidak relevan. Oleh karena itu, hanya dipakai fitur kognat dengan karakter yang berbeda, yaitu yang menggunakan pemetaan karakter.

Penanganan untuk masalah tersebut dilakukan dengan memetakan karakter dari bahasa Jepang (hiragana, katakana, dan Kanji) ke dalam karakter bahasa Indonesia (alfabet). Kakas yang menyediakan pemetaan tersebut banyak tersedia di internet. Salah satunya yang tersedia secara gratis adalah J-Talk[18].

III.3 Metode Evaluasi

Terdapat 2 evaluasi yang dilakukan dalam tugas akhir ini. Yang pertama adalah evaluasi model pembelajaran yang dibuat dengan mengambil data latih sebagai data uji (pengujian tertutup). Yang kedua adalah evaluasi kamus dwibahasa yang didapatkan dari korpus (pengujian terbuka). Kedua evaluasi tersebut dilakukan secara manual dan terbagi ke dalam 2 kasus berikut.

Kasus I. Evaluasi Entri Berupa Kata

1. Cari kata tersebut di dalam kamus umum bahasa Indonesia-Jepang;
2. Jika kedua kata ditemukan berpasangan, entri dinyatakan benar;
3. Jika tidak ditemukan, entri dinyatakan salah.

Kasus II. Evaluasi Entri Berupa Frasa

1. Cari frasa tersebut dengan melakukan *query* pada laman web Weblio[19];
2. Jika ditemukan kalimat yang mengandung frasa tersebut dalam bahasa lawannya, entri dinyatakan benar;
3. Jika tidak ditemukan, entri dinyatakan salah.

Selain itu, diadopsi juga sistem evaluasi manual yang dilakukan oleh Aker dkk.

[18] `http://nihongo.j-talk.com/`
[19] `http://ejje.weblio.jp/sentence/`

(2013) dengan membagi translasi menjadi 2 macam, yaitu translasi penuh dan translasi parsial. Translasi dikatakan penuh jika kedua kata/frasa adalah translasi yang benar. Translasi dikatakan parsial jika terdapat kata yang tidak memiliki translasi dalam bahasa lawannya, namun terdapat juga kata yang memiliki translasi. Translasi parsial hanya mungkin terjadi pada pasangan frasa.

Dalam kasus translasi parsial, pasangan frasa yang terekstrak dianggap sebagai entri yang benar. Oleh karena itu, pada akhir percobaan akan terdapat 2 ukuran, yaitu persentase translasi penuh (hanya menghitung translasi penuh sebagai entri yang benar) dan presentasi translasi parsial (menghitung translasi penuh dan parsial sebagai entri benar).

DAFTAR PUSTAKA

Aker, A., Paramita, M. L., & Gaizauskas, R. J. (2013). Extracting bilingual terminologies from comparable corpora. In *Acl (1)* (pp. 402–411).

Ananiadou, S. (1994). A methodology for automatic term recognition. In *Proceedings of the 15th conference on computational linguistics-volume 2* (pp. 1034–1038). Association for Computational Linguistics.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on computational learning theory* (pp. 144–152). ACM.

Cristianini, N. & Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge: Cambridge University Press.

Daille, B. & Morin, E. (2005). French-english terminology extraction from comparable corpora. In *Natural language processing–ijcnlp 2005* (pp. 707–718). Springer.

Fujii, A., Ishikawa, T., & Lee, J.-H. (2004). Term extraction from korean corpora via japanese. In *Proceedings of the 3rd international workshop on computational terminology* (pp. 71–74).

Fung, P. (1995). Compiling bilingual lexicon entries from a non-parallel english-chinese corpus. Proceedings of the 3rd Annual Workshop on Very Large Corpora.

Jagarlamudi, J. & Daumé III, H. (2010). Extracting multilingual topics from unaligned comparable corpora. In *Advances in information retrieval* (pp. 444–456). Springer.

Koehn, P. (2010). *Statistical machine translation*. New York: Cambridge University Press.

Lefever, E., Macken, L., & Hoste, V. (2009). Language-independent bilingual terminology extraction from a multilingual parallel corpus. In *Proceedings of the*

12th conference of the european chapter of the association for computational linguistics (pp. 496–504). Association for Computational Linguistics.

Limanthie, N. A. (2014). *Pembangunan kamus indonesia-jepang berdasarkan frekuensi pasangan kata dari wikipedia* (Bachelor's thesis, Institut Teknologi Bandung, Bandung: Program Studi Teknik Informatika).

Mitchell, T. M. (1997). *Machine learning*. New York: McGraw-Hill.

Rapp, R. (1995). Identifying word translations in non-parallel texts. In *Proceedings of the 33rd annual meeting on association for computational linguistics* (pp. 320–322). Association for Computational Linguistics.

Tiedemann, J. (2012, May 23–25). Parallel data, tools and interfaces in opus. In N. C. (Chair), K. Choukri, T. Declerck, M. U. Dogan, B. Maegaard, J. Mariani, ... S. Piperidis (Eds.), *Proceedings of the eight international conference on language resources and evaluation (lrec'12)*. Istanbul, Turkey: European Language Resources Association (ELRA).

Tsuji, K., Daille, B., & Kageura, K. (2002). Extracting french-japanese word pairs from bilingual corpora based on transliteration rules. In *Lrec*.

Vapnik, V. N. (2000). *The nature of statistical learning theory* (2nd). New York: Springer-Verlag.

Yu, K. & Tsujii, J. (2009). Bilingual dictionary extraction from wikipedia. In *Proceedings of machine translation summit xii* (pp. 379–386).

Lampiran A Posisi Tugas Akhir

A.1 Tabel Posisi

Pembangunan kamus dwibahasa termasuk salah satu penerapan dari ekstraksi istilah dwibahasa. Posisi tugas akhir dalam studi literatur diberikan pada Tabel A.1.

Tabel A.1. Posisi tugas akhir dalam studi literatur

Studi	**ATR**						**Bahasa**
	banyak bahasa		pendekatan		korpus		
	eka	dwi	ling.	stat.	par.	*comp.*	
Ananiadou (1994)	V		V		V	V	Inggris
Tsuji, Daille, dan Kageura (2002)		V	V		V	V	Jepang-Prancis
Fujii, Ishikawa, dan Lee (2004)		V	V		V	V	Jepang-Korea
Daille dan Morin (2005)		V		V		V	Inggris-Prancis
Lefever, Macken, dan Hoste (2009)		V		V	V		Belanda-Inggris-Italia-Prancis
Aker, Paramita, dan Gaizauskas (2013)		V	V	V		V	22 negara Eropa
Limanthie (2014)		V		V		V	Indonesia-Jepang
tugas akhir		V	V	V		V	Indonesia-Jepang

A.2 Anak Lampiran Lainnya

appendix appendix appendix appendix appendix appendix

Lampiran B Contoh Lampiran Lainnya

appendix appendix appendix appendix appendix appendix