

37473/H/10



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

RTIF  
005.45  
Aad  
p-1  
Jev 8

**TESIS C12541**  
**PENGEMBANGAN EXAMPLE BASED MACHINE**  
**TRANSLATION DENGAN SISTEM TASHRIF PADA**  
**SEBUAH COMPUTER AIDED TRANSLATION ARAB**  
**- INDONESIA**

**KHARISMAN KHOLID HUDAYA**  
5105 201 018

**DOSEN PEMBIMBING**  
Dr. H. Agus Zainal, S.Kom., M.Kom.

**PROGRAM MAGISTER**  
**BIDANG KEAHLIHAN TEKNIK INFORMATIKA**  
**JURUSAN TEKNIK INFORMATIKA**  
**FAKULTAS TEKNOLOGI INFORMASI**  
**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**  
**SURABAYA**  
**2008**

<b>PERPUSTAKAAN ITS</b>	
Tgl. Terima	13-8-2009
Terima Dari	H
ITS Agenda Prp.	21


**Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar  
Magister Komputer (M.Kom.)  
di  
Institut Teknologi Sepuluh Nopember**

**oleh:  
KHARISMAN KHOLID HUDAYA  
NRP. 5105 201 018**


**Tanggal Ujian: 29 Juli 2008  
Periode Wisuda: September 2009**

**Disetujui oleh:**

**1. Dr. Agus Zaenal Arifin, S.Kom, M.Kom  
NIP. 132 137 825**

  
.....  
( **Pembimbing** )

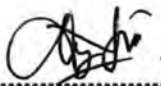
**2. Victor Hariadi, S.Si, M.Kom  
NIP. 132 125 657**

  
.....  
( **Penguji** )

**3. Chastine Fatichah, S.Kom, M.Kom  
NIP. 132 298 829**

  
.....  
( **Penguji** )

**4. Anny Yuniarti, S.Kom, M.Sc  
NIP. 132 309 746**

  
.....  
( **Penguji** )

  
**Direktur Program Pascasarjana,**  
**Prof. Dr. Ir. Suparno, MSIE**  
**NIP. 130 532 035**

## **Pengembangan Example Based Machine Translation dengan Sistem Tasrif pada Sebuah Computer Aided Translation Arab-Indonesia**

**Nama Mahasiswa : Kharisman Kholid Hudaya**  
**NRP : 5105201018**  
**Pembimbing : Dr. H. Agus Zainal, S.Kom, M.Kom**

### **Abstrak**

Example Based Machine Translation (EBMT) adalah sebuah paradigma mesin penerjemahan yang menggunakan sumber corpus contoh terjemahan. Paradigma ini lebih baru dibandingkan dua paradigma lain: *Rule Based Machine Translation* dan *Statistical Machine Translation*. Tingkat akurasi output terjemahan dari EBMT tergantung dari kelengkapan cakupan corpus contoh terjemahan. Kemampuan EBMT dibatasi dengan jumlah dan jangkauan dari training datanya. Bahkan dengan corpus yang sangat besar, kita tidak akan memiliki contoh terjemahan yang mencakup segala kemungkinan kalimat yang ingin diterjemahkan.

Kualitas hasil terjemahan suatu kalimat yang dihasilkan mesin penerjemah seringkali jauh dari harapan. Karena itu, penerjemahan dengan bantuan computer (Computer Aided Translation) lebih dipilih, karena manusia memiliki lebih banyak kontrol dalam mengklasifikasikan suatu kata berdasarkan makna kontekstualnya. Penelitian ini bertujuan mengembangkan suatu prototipe Computer Aided Translation Arab – Indonesia yang mengadaptasi paradigma EBMT, ditunjang dengan Sistem Tashrif (*Tashrif System*), sebuah metode baru yang diperkenalkan. Ujicoba terhadap literatur berbahasa Arab dalam domain Ilmu Fiqh menunjukkan presisi 71% sedangkan pada portal berita online Al-Jazeera menunjukkan presisi 42%.

**Key words :** EBMT, Computer Aided Translation, Arab–Indonesia, Machine Translation.

**Improving Example Based Machine Translation with Tashrif  
System on a  
Computer-Aided Translation Arabic-Indonesia**

**By : Kharisman Kholid Hudaya**  
**Student Identity Number : 5105201018**  
**Supervisor : Dr. H. Agus Zainal, S.Kom, M.Kom**

**Abstract**

*Example Based Machine Translation (EBMT)* is a translation paradigm that use example translation corpus source. This paradigm is the newest than two paradigm exist before: *Rule Based Machine Translation* and *Statistical Machine Translation*. The accuracy of output translation from EBMT is depend on completeness coverage example corpus translation. Even with a reasonably large corpus, we will not have examples that cover everything we want to translate.

The quality of ouput sentence produced by machine translation often out of expectation. Therefore, Computer Aided Translation more eligible because human has more control to classify a word to a contextual meaning. This research had a purpose to develop a Computer Aided Translation Arab-Indonesia prototype with EBMT paradigm adaptation, extended by Tashrif System as a proposed new method. Experiment on domain Fiqh Islam get a total average precision 73% and 42% on Arabic news website Al-Jazeera.

**Key words :** EBMT, Computer Aided Translation, Arab-Indonesia, Machine Translation

## KATA PENGANTAR

Alhamdulillah, segala puji bagi Allah Subhaanahu Wa Ta'ala satu-satunya Dzat Yang berhak disembah. Hanya dengan perkenan dan pertolonganNya penulis dapat menyelesaikan tesis ini. Semoga sholawat dan salam senantiasa tercurah kepada junjungan penulis Nabi Muhammad Shollallaahu 'Alaihi Wasallam, keluarga, para Sahabat dan segenap yang mengikuti Sunnah beliau dengan baik.

Bagi sebagian orang, menyelesaikan tugas akhir study adalah sebuah perjuangan yang tak kan terlupakan. Demikianlah pula yang penulis rasakan saat menyelesaikan tesis ini. Segala rintangan yang ada sempat dianggap sebagai sandungan yang akan menjadi akhir perjuangan. Pernik-pernik pergulatan melawan kemalasan akibat kurang motivasi, kurang ulet dan disiplin, mestinya tidak sekedar dikenang, namun disisakan sebagai pelajaran.

Kurangnya tawakkal dan keyakinan kepada ArRahmaan (Dzat Yang Maha Penyayang) bisa disebut memiliki peran yang tak ringan. Bukankah Dia Yang Maha Pengasih telah memberi banyak fasilitas dan kemudahan? Sebagai bagian dari janjiNya: setiap satu kesulitan akan dibarengi dengan 2 kemudahan (Surat Alam Nasyrah ayat 5-6). Kalaulah tidak karena perkenanNya, tidak ada yang bisa penulis selesaikan. Dialah satu-satunya tempat bersandar dan menyerahkan segenap persoalan.

Penulis juga ingin mengucapkan terima kasih yang tulus kepada:

1. Dosen Pembimbing, Bpk Dr. H. Agus Zainal, S.Kom, M.Kom, pembimbing akademik terbaik yang pernah penulis temui, atas segala kesabaran dan ketelatenannya membimbing penulis hingga mampu menyelesaikan tesis ini
2. Ayahanda H. Sudirman Rais, SH, MM dan Ibunda Mintarsih Sudirman atas segenap pengorbanan, kasih sayang, dan segala yang tak kan mampu terbalas
3. Kakak penulis Eva Early, ST dan Abdi Sofyan, ST atas segenap dukungan yang telah diberikan
4. Nurul Sofyani, istri tercinta yang sukses 'menghambat' terselesaikannya tesis ini. Terima kasih atas 'hambatan'- 'hambatan' yang menyenangkan dan mencerahkan.

5. Huzaini, S.Kom rekan kerja penulis yang harus berjibaku menyelesaikan tugas-tugas Sekretariat PKH (Program Keluarga Harapan) yang sering penulis tinggalkan karena sibuk menyelesaikan tesis ini.
6. Semua pihak yang telah membantu terselesaikannya tesis ini yang tidak bisa penulis sebutkan satu persatu.

Penulis menyadari bahwa tesis ini bertabur banyak kekurangan, terlalu jauh dari kesempurnaan. Hal itu bertautan erat dengan kekurangsempurnaan penulisnya. Kritik dan saran yang membangun tetap diharapkan guna pengembangan yang bermanfaat bagi sesama.

## DAFTAR GAMBAR

Gambar 2.1	Template Huruf Pembentuk Kata Kerja.....	14
Gambar 3.1	Diagram Alur Utama Proses Penerjemahan.....	40
Gambar 3.2	Kode VB6 untuk menghilangkan harakat pada kata.....	42
Gambar 3.3	Keterhubungan tabel Contoh dan Word_alignment.....	45
Gambar 3.4	Fungsi Pemecah Kalimat Menjadi Kata Penyusunnya.....	49
Gambar 3.5	Potongan Kode Visual Basic 6 untuk Penyesuaian Partikel Awalan Kata Benda.....	56
Gambar 4.1	Skema Tashrif System.....	57
Gambar 4.2	Kode Visual Basic 6 untuk Mendeteksi <i>Fi'il Mudhari'</i> .....	58
Gambar 5.1	Screenshot Tampilan Petunjuk Kata per Kata.....	73

## DAFTAR TABEL

Tabel 2.1	Kosa Kata Bahasa Arab yang bermakna 'kuda'.....	8
Tabel 2.2	Perbedaan Jabatan Kata 'Haamid' dalam kalimat.....	9
Tabel 2.3	Perubahan Tanda Baca karena Perbedaan I'rab.....	9
Tabel 2.4	Contoh kata yang tersusun dari huruf 'kaaf, ta', dan ba'.....	10
Tabel 2.5	Kata Benda Penunjuk Waktu dengan Perasaan Kebahasaan yang Detail.....	12
Tabel 2.6	Daftar wazan.....	15
Tabel 2.7	Jenis Kata dalam Tashrif .....	16
Tabel 2.8	Contoh Penempatan Kata كَتَبَ pada masing-masing jenis kata.	18
Tabel 2.9	<i>Dhomir</i> (kata ganti personal) dalam <i>tashrif</i> .....	19
Tabel 2.10	<i>Tashrif Lughowy</i> untuk fi'il madhi كَتَبَ .....	20
Tabel 2.11	Jumlah Huruf Pembentuk Kata.....	20
Tabel 2.12	Macam-macam Bangunan Kata ( <i>bina'</i> ) dalam <i>tashrif</i> .....	22
Tabel 2.13	Contoh Peletakan Template pada Literatur <i>Al-Amsilatut Tashriyyah</i> .....	23
Tabel 2.14	Skor Kecocokan pada Tiap Level.....	32
Tabel 2.15	<i>Alignment LookUp Table</i> .....	34
Tabel 3.1	Contoh <i>LookUp Alignment</i> Masing-masing Kata Inputan.....	41
Tabel 3.2	Contoh Penyimpanan Kata Benda, Terjemahan, dan <i>Index Non Harakat</i> .....	42
Tabel 3.3	Struktur Tabel Contoh.....	44
Tabel 3.4	Struktur Tabel <i>Word_alignment</i> .....	45
Tabel 3.5	Struktur Tabel Jama sebagai Penyimpan Daftar Kata Benda....	46
Tabel 3.6	Struktur Tabel Induk1 sebagai List Kata Kerja.....	46
Tabel 3.7	Struktur Tabel Sinonim.....	47
Tabel 3.8	Posisi Huruf Pembentuk <i>Fi'il Madhi bina' shohih</i> pada Template Wazan.....	50



Tabel 3.9	Penyesuaian Struktur Tabel di Database untuk Rule Huruf Pembentuk <i>Fi'il Madhi</i> .....	51
Tabel 3.10	Contoh Isi Data pada Tabel 'Contoh'.....	52
Tabel 3.11	Isi Tabel Word_alignment untuk ID_KAL 11.....	53
Tabel 3.12	Contoh Isi Data Tabel Jama.....	54
Tabel 3.13	Contoh Isian Data di Tabel Induk1 (List Kata Kerja).....	55
Tabel 4.1	Perbedaan Contoh Template pada Literatur antara <i>Mashdar Mim</i> dengan <i>Isim Zamaan</i> .....	60
Tabel 4.2	Peletakan Template Berbasis Pola pada Tashrif System.....	60
Tabel 4.3	Pengubahan Orientasi Baris dan Kolom.....	61
Tabel 4.4	Peletakan Pola Kata dengan Kode.....	62
Tabel 4.5	Peletakan Pola Template dengan Tambahan Kolom Kd_Dhamir.....	63
Tabel 5.1	Ujicoba Pengentryan satu kata pada training data.....	69
Tabel 5.2	Pengujian Inputan Kalimat/ Frase pada <i>Training Data</i> .....	72
Tabel 5.3	Pengujian terhadap Data Tes pada Domain Ilmu Fiqh.....	73
Tabel 5.4	Pengujian terhadap Data Tes pada Situs Berita.....	73
Tabel 5.5	Nilai Pengujian terhadap Data Tes pada 'الملخص الفقهي'.....	74
Tabel 5.6	Hasil Pengujian terhadap Situs Berita Online Al-Jazeera.....	75

**DAFTAR ISI**

## DAFTAR ISI

Lembar Persetujuan.....	i
Abstrak Indonesia .....	iii
Abstrak Inggris.....	v
Kata Pengantar.....	vii
Daftar Gambar.....	ix
Daftar Tabel .....	xi
Daftar Isi.....	xiii
<b>BAB 1 PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah .....	3
1.3 Tujuan, Kontribusi, dan Manfaat Penelitian.....	3
1.4 Metode Penelitian.....	4
1.5 Sistematika Pembahasan.....	6
<b>BAB 2 TINJAUAN PUSTAKA.....</b>	<b>7</b>
2.1 Ilmu Bahasa Arab.....	7
2.1.1 Keunikan dan Keistimewaan Bahasa Arab.....	7
2.1.2 Ilmu Tashrif dalam Struktur Bahasa Arab.....	11
2.2 Penjelasan tentang Machine Translation.....	23
2.3 Example Based Machine Translation Arab-English.....	29
<b>BAB 3 COMPUTER AIDED TRANSLATION</b>	
<b>ARAB-INDONESIA.....</b>	<b>39</b>
3.1 Pengertian Computer Aided Translation (CAT).....	39
3.2 Perancangan CAT Arab-Indonesia	
3.2.1 Alur Utama Proses Penerjemahan.....	39
3.2.2 Dukungan yang Diharapkan dari Sistem.....	40
3.2.3 Menangani Input tanpa Harakat .....	41
3.2.4 Menyiapkan Struktur Database.....	43

3.2.5	Struktur Tabel dalam Database.....	44
3.2.6	Proses Memecah Kalimat Menjadi Kata-kata Penyusunnya.....	48
3.2.7	Ekstraksi Huruf Penyusun pada Kata Kerja.....	49
3.2.8	Proses Pencocokan Input Kata dengan Database.....	52
<b>BAB 4</b>	<b>TASHRIF SYSTEM.....</b>	<b>57</b>
4.1	Pendahuluan.....	57
4.2	Subsistem Analisis.....	58
4.3	Subsistem Sintesis.....	59
4.3.1	Penyesuaian Pola.....	59
4.3.2	Penyesuaian Harakat pada Mashdar dan Mashdar Mim...	59
4.3.3	Pengubahan Orientasi dari Kolom ke Baris.....	61
<b>BAB 5</b>	<b>UJI COBA DAN PENGEMBANGAN LEBIH LANJUT....</b>	<b>65</b>
5.1	Data.....	65
5.1.1	Data Inputan.....	65
5.1.2	<i>Training Data</i> .....	65
5.1.3	<i>Testing Data</i> (Data Tes) .....	67
5.2	Skenario Percobaan.....	67
5.2.1	Skenario I : Percobaan terhadap <i>Training Data</i> .....	67
5.2.2	Skenario II : Percobaan terhadap <i>Testing Data</i> .....	68
5.3	Pengujian terhadap <i>Training Data</i> .....	68
5.3.1	Pengujian 1 Kata.....	68
5.3.2	Pengujian Kalimat/ Frase.....	70
5.4	Pengujian terhadap <i>Testing Data</i> .....	71
5.4.1	Pengujian terhadap Data Tes pada Domain Ilmu Fiqh (الملخص الفقهي).....	71
5.4.2	Pengujian terhadap Data Tes pada Situs Portal Berita Al-Jazeera.....	73

5.5	Analisis dan Evaluasi.....	74
5.6	Kesimpulan.....	75
5.7	Pengembangan Lebih Lanjut.....	76
DAFTAR PUSTAKA.....		77



**BAB 1**  
**PENDAHULUAN**

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Bahasa Arab adalah salah satu dari tujuh bahasa resmi yang digunakan UNO (*United Nations Organization*) dan merupakan bahasa ibu (*mother tongue*) dari sekitar 300 juta orang (Ethnologue, 2008). Bagi umat Islam, bahasa Arab memiliki nilai tersendiri karena merupakan bahasa kitab sucinya (AlQur'an) dan juga bahasa al-Hadits (sabda Rasulullah *shollallaahu 'alaihi wasallam*).

Penggunaan komputer sebagai media bantu manusia khususnya dalam hal pemrosesan bahasa alami (*natural language processing*) telah banyak diupayakan. Diantaranya dengan dikembangkannya *machine readable dictionary* (kamus elektronis) ataupun *machine translation* (penerjemah elektronis). Dengan media bantu elektronis tersebut diharapkan manusia lebih mudah dalam menerjemahkan suatu bahasa alami ke bahasa alami yang lain.

Pemrosesan bahasa Arab dalam ilmu komputer menjadi topik yang menarik karena sifat bahasa tersebut yang unik. Keunikan bahasa Arab tersebut di antaranya :

1. Penggunaan abjad / alphabet yang berbeda dengan mayoritas bahasa lain.
2. Orientasi teks dari kanan ke kiri (*right to left*).
3. Naskah – naskah yang banyak tertulis tanpa harakat / tanpa huruf vokal (*unvowelized form*).
4. *Non concatenative language*. Perubahan subyek dan predikat sudah terkait dengan 'bentuk permukaan' (*surface form*) pada kata, bukan dengan menggandengkan subyek dan predikat yang berbeda pada kata yang sudah ada.
5. Morfologi yang kompleks.
6. *Highly productive : derivationally and inflectionally*. (William J. Black, 2004)

Terdapat banyak penelitian tentang *machine translation* untuk Bahasa Arab ke bahasa lainnya, khususnya dari bahasa Arab ke bahasa Inggris dan sebaliknya. Namun, belum dijumpai adanya *machine translation* yang mengakomodasi

penerjemahan dari bahasa Arab ke bahasa Indonesia maupun sebaliknya. Software-software yang sudah ada masih berkisar pada kamus elektronik yang bersifat menerjemahkan satu kata ke satu kata yang lain (*word by word*), bukan berupa penerjemah elektronik yang mampu menerjemahkan satu kalimat atau frase. Di antara software kamus elektronik Arab-Indonesia (*reversible*) yang tersedia saat ini adalah *al-Mufiid*. Namun, software tersebut masih belum bisa mengakomodasi perubahan morfologi kata dalam bahasa Arab secara otomatis. Semua pencocokan kata adalah berdasarkan kecocokan yang tepat sama (*exactly match*).

Pendekatan dalam merancang sebuah *machine translation* terdiri dari : (i) *Rule-Based Machine Translation* (RBMT) , (ii) *Statistical Machine Translation* (SMT) , dan (iii) *Example-Based Machine Translation* (EBMT). Pada RBMT sumber pengetahuan dari *machine* adalah sekumpulan aturan-aturan (*rules*) yang telah didefinisikan. Sedangkan pada SMT pencocokan kata antara bahasa sumber dengan bahasa target adalah berdasarkan statistik pemakaian dan pilihan yang cocok sebelumnya. EBMT menekankan pada contoh-contoh penerjemahan dalam satu paragraph atau satu kalimat antara bahasa sumber dengan bahasa target yang telah ada sebelumnya.

EBMT adalah metode yang terbaru dibandingkan kedua metode lainnya. Kemunculannya juga terjadi sebagai solusi atas kekurangan yang ada pada RBMT dan SMT. Namun, EBMT tidak mungkin diterapkan secara sendirian. Hal ini disebabkan karena tidak akan pernah ada suatu *machine translation* yang memiliki basis data berupa contoh-contoh penerjemahan antar bahasa yang mampu menangani seluruh kemungkinan input frase atau kalimat. Apalagi untuk bahasa Arab yang merupakan bahasa yang morfologinya sangat kompleks dan bervariasi tinggi.

Karena itu, seharusnya sebuah *Arabic Machine Translation* yang baik tidak bisa mengandalkan satu pendekatan saja. Lebih tepatnya digunakan *hybrid system*, atau pendekatan EBMT –sebagai pendekatan terbaik- yang dikembangkan dengan fungsionalitas / aturan-aturan tambahan. Aturan-aturan tersebut diperlukan untuk mengakomodasi input kalimat yang tidak ada contohnya pada basis data *knowledge system*.



## 1.2 Rumusan Masalah

Berdasarkan penjelasan tersebut di atas, masalah-masalah dalam penelitian ini dapat dirumuskan sebagai berikut :

1. Bagaimana membangun sebuah modul EBMT Arab –Indonesia yang adaptif, yang mampu mengakomodasi penerjemahan pada untaian kalimat yang tidak terdapat pada contoh terjemahan sebelumnya ?
2. Bagaimana membuat sebuah *engine* penganalisis morfologi kata dalam bahasa Arab yang mengakomodasi penerapan ilmu *tashrif* (pada tata bahasa Arab) ?

## 1.3 Tujuan dan Manfaat Penelitian (Kontribusi)

Penelitian ini dibuat untuk menyusun sebuah prototipe penganalisa morfologi kata (*morphological analyzer*) sebagai salah satu modul/ komponen dasar dari pembuatan *Arabic – Indonesian machine translation* yang lebih kompleks. *Morphological analyzer* yang akan dibuat ini benar-benar mendasarkan pada ilmu analisis morfologi dalam dalam tata bahasa Arab, karena itulah disebut sebagai *Tashrif System*.

Dari penelitian ini diharapkan bisa dihasilkan sebuah software kamus elektronis Arab-Indonesia dengan kemampuan menelaah perubahan morfologi kata yang bisa menerima input kata tanpa *harakat (non-vowelization)*. Sebelum penelitian ini dirancang, belum ditemukan adanya software kamus elektronis Arab-Indonesia dengan kemampuan tersebut. Software yang ada semacam al-Mufid 1.0 hanya bisa mencari terjemahan kata dengan metode *exact match* (persis sama), dan tidak bisa mendeteksi terjemahan kata dari suatu kata yang akar katanya sama, namun tidak ada dalam database. Dengan prototype yang akan dihasilkan ini diharapkan akan semakin berkembang pengkajian terhadap literatur –literatur berbahasa Arab khususnya kitab-kitab referensi religius Islam. Software yang akan dirancang ini ditujukan untuk para pelajar dan mahasiswa dengan kemampuan komputer menengah, yang sudah terbiasa mengoperasikan aplikasi pengolah kata (*word processor*) standart.

Secara umum, kontribusi dari penelitian ini adalah :

1. Pengenalan dan penjelasan tentang metode baru : *Tashrif system*. Sebuah metode analisis morfologi kata dalam bahasa arab yang berbasiskan *template/wazan* yang telah didefinisikan sebelumnya, dengan seperangkat aturan (*rule*) dan algoritma untuk mencococokkan suatu kata dengan *template* dan aturan-aturan tersebut. Metode tersebut diinspirasi oleh penyusunan *template – template* bangunan kata bahasa Arab yang terdapat dalam buku *al-Amtsilatut Tashrifityyah*(Maksum bin Ali, 1965).
2. Menawarkan sebuah alternatif pengembangan metode *Example – Based Machine Translation* (EBMT). Dalam penelitian ini, akan dijelaskan salah satu alternatif pengembangan sebuah EBMT jika tidak didapati adanya contoh terjemahan sebelumnya yang memenuhi syarat.

#### 1.4. Metode Penelitian

Dalam menyusun penelitian ini, dilakukan langkah-langkah sebagai berikut :

##### 1. Studi literatur

Dilakukan studi literatur terhadap penelitian-penelitian sebelumnya tentang : *machine translation* secara umum, *Example Based Machine Translation*, *arabic stemming*, *arabic linguistics*, *morphological analyzer*, *part-of-speech tagger*.

##### 2. Memasukkan contoh terjemahan Bahasa Arab - Indonesia ke dalam DBMS .

Untuk mendapatkan contoh terjemahan, dilakukan pengumpulan artikel / e-book Islamy yang merupakan terjemahan dari kitab asli berbahasa Arab ke bahasa Indonesia. Setelah didownload suatu file, kemudian dicarilah artikel aslinya dalam bahasa Arab. Pencarian naskah asli berbahasa Arab tersebut bisa didapatkan dari Software *Maktabah asy-Syaamila* atau mencarinya pada situs-situs penyedia layanan naskah asli berbahasa arab semisal : [www.sahab.net](http://www.sahab.net), Setelah ditemukan naskah asli dan terjemahannya yang sesuai, dimasukkanlah teks-nya ke dalam DBMS. Untuk menampung data tersebut, dibuatlah sebuah tabel dengan 2 field : Indonesia dan Arabic. Setiap recordnya diisi dengan sebuah kalimat.

##### 3. Membuat list dictionary per kata.

Sumber datanya adalah Kamus Arab-Indonesia al-*Munawwir*.

4. Mendesain modul software yang mengakomodasi *Example Based Machine Translation (EBMT)*.

Modul untuk menjalankan EBMT dirancang dengan kemampuan menjalankan tahapan :

a. *Matching*

Mencari padanan *fragment* frase pada berbahasa Arab pada database contoh terjemahan yang sesuai dengan fragment frase pada input kalimat. Pencarian tersebut dilakukan dengan menggunakan query SQL. Fragment frase pada contoh terjemahan yang sesuai adalah yang memenuhi paling tidak 2 kata berdampingan.

b. *Scoring*

Kumpulan fragment frase yang lolos pada tahapan matching diberi bobot sesuai dengan ketepatannya. Ketepatan tersebut meliputi : ketepatan persis sama huruf per huruf (*exactly match*) dan kesamaan *stem*. Untuk mempersempit jumlah data yang lolos pada tahapan selanjutnya ditetapkan nilai *threshold* tertentu.

c. *Transfer dan Rekombinasi*

Mencari padanan kata dalam bahasa Inggris dari fragment frase berbahasa Arab yang memiliki *score* yang paling tinggi, kemudian menggabungkan fragment terjemahan tersebut.

5. Mendesain modul *morphological analyzer* untuk menangani input kata yang fragment frase-nya tidak cocok dengan contoh terjemahan. Dalam penelitian ini, *morphological analyzer* yang dirancang adalah untuk menganalisis 1 kata inputan. Untuk 1 kata inputan, tahapan-tahapan yang dilakukan:

a. *Stemming*.

Dilakukan proses menghilangkan seluruh prefix dan posfix yang ada pada kata tersebut, sehingga didapatkan stem yang sesuai.

b. Pencarian kata dalam kamus.

c. Pencocokan kata dengan template yang telah ada.

6. Ujicoba dan Evaluasi pengembangan sistem serta menarik kesimpulan dari implementasinya.

### 1.5 Sistematika Pembahasan

Pada bab 2 dijelaskan tentang Bahasa Arab, keistimewaan Bahasa Arab, struktur bahasa yang berkaitan dengan tashrif, dan dijelaskan pula tentang *Machine Translation*, *Example Based Machine Translation* dan penelitian sebelumnya terkait *Example Based Machine Translation* Arab – Inggris.

Selanjutnya, pembahasan tentang *tashrif system* sebagai metode yang diperkenalkan, disebutkan dalam bab 3. Dijelaskan metode penyusunan template sebagai bentuk akomodasi dan penyesuaian dari buku literatur *al-Amtsilatut Tashrifiyah*, struktur tabel dalam DBMS, dan algoritma mengekstraksi template tersebut.

Pada bab 4 dijelaskan tentang bagaimana penyesuaian konsep *Example Based Machine Translation* agar bisa memberikan dukungan yang optimal terhadap sebuah *Computer Aided Translation* Arab – Indonesia. Dijelaskan pula tentang struktur tabel DBMS dalam menyimpan contoh terjemahan, algoritma ekstraksi stem, dan integrasinya dengan *tashrif system*.

Terakhir, pada bab 5 dijelaskan tentang ujicoba yang telah dilakukan, hasil, kesimpulan, serta pengembangan lebih lanjut.

**BAB 2**  
**TINJAUAN PUSTAKA**

## BAB 2

### TINJAUAN PUSTAKA

#### 2.1 Ilmu Bahasa Arab

Bahasa Arab memiliki 2 jenis kata : *mudzakkar* (laki-laki), dan *muannats* (wanita). Berdasarkan jumlahnya, kata dalam bahasa Arab dibedakan menjadi : *mufrad* (tunggal), *mitsannaa* (dua), dan *jama'* (lebih dari 2). Suatu kalimat atau frase dalam bahasa Arab akan tersusun dari tiga unsur pembentuknya. Ketiga macam itu adalah kata benda (*isim*), kata kerja (*fi'il*), dan huruf. *Isim* adalah kata yang menunjukkan pada sesuatu namun tidak terikat dengan waktu, sedangkan *fi'il* adalah kata yang menunjukkan sesuatu pekerjaan yang terikat dengan waktu tertentu (telah selesai, sedang berlangsung, atau akan dilakukan). Huruf sebagai unsur pembentuk suatu kata, yang dimaksud bukanlah huruf hijaiyyah. Unsur pembentuk kata ini dibedakan dari *isim* dan *fi'il* karena tidak memiliki karakteristik seperti dua unsur pembentuk tersebut (Umam, 2004).

Secara gramatikal, kata benda dalam bahasa Arab bisa mengalami keadaan : *rofa'*, *nashob*, dan *jar*. Contoh keadaan *rofa'* dari suatu kata benda adalah pada saat berkedudukan sebagai subyek, sedangkan *nashob* misalnya pada saat jabatan kata tersebut adalah sebagai obyek penderita. Keadaan *jar* salah satunya dialami suatu kata jika didahului oleh sebuah preposisi. Kata benda dalam bahasa Arab juga dibedakan menjadi kata benda tertentu (*ma'rifah*), dan umum (*nakirah*) (Umam, 2004).

#### 2.1.1 Keunikan dan Keistimewaan Bahasa Arab

Bahasa Arab memiliki keunikan dan keistimewaan dibandingkan bahasa lain. Beberapa poin penting sebagaimana dijelaskan dalam (Azhar, 2007) bisa disimak sebagai berikut :

1. Bahasa Arab adalah Bahasa yang Kaya dengan Kosa Kata

Dalam bahasa Arab, satu kata bisa memiliki perbendaharaan makna yang cukup banyak. Misalkan, perkataan *'ain* bisa berarti : mata penglihatan, mata air, sebuah negeri, sebuah tempat, sebuah kaum, komandan pasukan,

makna diri, pembayaran kontan, pengintip, dan huruf 'ain. Selain itu, untuk merujuk pada suatu makna tertentu dengan perasaan kebahasaan yang detail terdapat kosa kata yang khusus. Sebagai contoh, kosa kata yang menunjukkan berbagai kategori kuda, diungkapkan dalam tabel 2.1:

Tabel 2.1 Kosa Kata Bahasa Arab yang bermakna 'kuda'

Kosa Kata	Makna
خَيْلٌ	sekumpulan kuda
فَرَسٌ	seekor kuda (jantan atau betina)
حِصَانٌ	kuda jantan
حَجْرٌ	kuda betina
مَهْرٌ	Anak kuda jantan
مَهْرَةٌ	Anak kuda betina
فِلْوٌ	anak kuda jantan yang baru lepas dari menyusui
هَيْكَلٌ	kuda besar bertubuh tegap
مَطْهَمٌ	Kuda yang sempurna dan baik

## 2. Bahasa Arab Mempunyai Kaidah Analisis Struktur kata (*i'rab*) yang Sempurna

Analisis ini digambarkan dalam bentuk perubahan baris-baris akhir perkataan akibat perbuahan struktur frase atau fungsi kata tersebut. Perubahan-perubahan *i'rab* tersebut akan memberikan kesan pada perubahan maksud perkataan dalam suatu susunan kalimat. Analisis bahasa dan perubahan struktur frase ini tidak terdapat dalam bahasa lain.

Contohnya, perkataan "*Haamid*" (nama orang laki-laki) dalam frase tertentu bisa dibaca dengan *Haamidun*, *Haamidani*, dan *Haamidin* karena berada dalam struktur dan pola kalimat yang berbeda kedudukannya seperti kalimat-kalimat dalam Tabel 2.2

Tabel 2.2 Perbedaan Jabatan Kata 'Haamid' dalam kalimat

Fraser	Makna
جَاءَ حَامِدٌ	Telah datang Hamid
رَأَيْتُ حَامِدًا	Aku melihat Hamid
مَرَرْتُ بِحَامِدٍ	Aku berpapasan dengan Hamid

Kata 'Hamid' dalam struktur frase pertama dibaca *Haamidun* berada dalam pola nominatif (*marfu'*) karena merupakan subyek kalimat. Dalam struktur frase kedua, kata tersebut dibaca *Haamidan* , berada dalam pola akusatif (*mansub*) karena merupakan obyek penderita (*maf'uul bih*). Sedangkan dalam frase ketiga dibaca *Haamidin* karena ; dalam pola genitif (*majrur*).

Begitu juga dengan penggunaan kata yang berlainan jenis seperti laki-laki (*mudzakkar*) dan perempuan (*muannats*) serta penggunaan kata tunggal (*mufrad*) , dua (*mutsanna*), dan lebih dari dua (*jama'*), dalam suatu susunan frase bisa membawa pada perubahan tanda baca, pembentukan kata, dan sistem *i'rabnya* (analisis jabatan kata). Contohnya seperti dalam Tabel 2.3.

Tabel 2.3 Perubahan Tanda Baca karena Perbedaan *i'rab*.

Kalimat	Makna
التَّلْمِيذُ يَذْهَبُ إِلَى الْمَدْرَسَةِ	Pelajar (lelaki) itu pergi ke sekolah
التَّلْمِيذَةُ تَذْهَبُ إِلَى الْمَدْرَسَةِ	pelajar (perempuan) itu pergi ke sekolah
التَّلْمِيذَانِ يَذْهَبَانِ إِلَى الْمَدْرَسَةِ	dua orang pelajar (lelaki) itu pergi ke sekolah
التَّلْمِيذَاتَانِ تَذْهَبَانِ إِلَى الْمَدْرَسَةِ	dua orang pelajar (perempuan) itu pergi ke sekolah
التَّلَامِيذُ يَذْهَبُونَ إِلَى الْمَدْرَسَةِ	pelajar-pelajar (lelaki) itu pergi ke sekolah
التَّلْمِيذَاتُ يَذْهَبْنَ إِلَى الْمَدْرَسَةِ	pelajar-pelajar (perempuan) itu pergi ke sekolah



Jika diperhatikan, enam struktur frase tersebut sebenarnya membawa maksud yang sama yaitu : "*pelajar itu pergi ke sekolah*", namun karena bilangan pelajar dan jenisnya yang berbeda menyebabkan perubahan pada sudut pembentukan kata, baris, tanda *i'rab* pada kata "pelajar" dan kata kerja "pergi".

### 3. Bahasa Arab Memiliki Morfologi yang Unik

Mayoritas kata dalam bahasa Arab terdiri dari 3 huruf inti. Tiga huruf ini menentukan kesamaan akar kata dari kata yang ada. Tiga huruf utama itu, biasanya disebut sebagai fa' , 'ain, dan laam fi'il. Kombinasi dari 3 huruf inilah yang menentukan morfologi dari suatu kata dalam bahasa Arab. Sebagai contoh, dari huruf utama kaaf, ta' dan ba' bisa terangkai menjadi kata-kata dalam Tabel 2.4:

Tabel 2.4 Contoh kata yang tersusun dari huruf 'kaaf', ta', dan ba'

Kata	Arti
كَبَّ	telah menulis
كِتَابٌ	Buku bacaan
مَكْتَبَةٌ	perpustakaan
كِتَابَةٌ	Penulisan

### 4. Bahasa Arab adalah Bahasa Berdaya Tahan dan Ringkas

AlQur'an yang berbahasa Arab banyak menggunakan ayat yang ringkas dan tepat untuk menunjukkan pada maksud tertentu. Teknik yang digunakan diantaranya membuang beberapa perkataan tertentu tanpa mengubah maksud yang ingin disampaikan. Sebagai contoh, disebutkan dalam al-Qur'an :

وَأَسْأَلِ الْقَرْيَةَ الَّتِي كُنَّا فِيهَا

"dan tanyakanlah kepada (penduduk) negeri yang kami telah tinggal di dalamnya (Q.S Yusuf : 82).

Dalam ayat tersebut, ada kata yang secara naluri kebahasaan terasa dihilangkan, yaitu أهل yang bermakna penduduk. Namun, penghilangan kata tersebut tidaklah mengurangi maksud yang terpahami dari ayat tersebut.

Demikian juga, banyak ayat yang ringkas dan padat yang cukup untuk menyampaikan suatu maksud tertentu yang sebenarnya jika diutarakan dengan bahasa lain akan panjang lebar. Sebagai contoh ayat berikut :

مَثَلُ الَّذِينَ حُمِّلُوا التَّوْرَةَ ثُمَّ لَمْ يَحْمِلُوهَا كَمَثَلِ الْحِمَارِ يَحْمِلُ أَسْفَارًا بِئْسَ

مَثَلُ الْقَوْمِ الَّذِينَ كَذَّبُوا بِآيَاتِ اللَّهِ وَاللَّهُ لَا يَهْدِي الْقَوْمَ الظَّالِمِينَ

*Perumpamaan orang-orang yang dipikulkan kepadanya Taurat, kemudian mereka tiada memikulnya adalah seperti keledai yang membawa kitab-kitab yang tebal. Amatlah buruk perumpamaan kaum yang mendustakan ayat-ayat Allah itu. Dan Allah tiada memberi petunjuk kepada kaum yang zalim (Q.S al-Jumua : 5).*

#### 5. Bahasa Arab Mempunyai Ungkapan yang Halus dan Teliti

Sebagai contoh, untuk mengungkapkan sebuah keadaan waktu, ada banyak kosa kata dengan penekanan perasaan bahasa yang detail, sebagaimana ditunjukkan pada Tabel 2.5. Secara umum, dalam bahasa Indonesia kita akan mengenal 3 jenis waktu utama, yaitu pagi, siang, dan sore. Pada bahasa Arab, ke-3 jenis waktu ini masih terklasifikasi lagi secara lebih detail. Pada waktu pagi saja ada 4 keadaan: دُرُورٌ, بَرُوعٌ, ضُحَىٰ dan غَزَالَةٌ. Waktu siang terbagi menjadi 3 keadaan: زَوَالٌ, ظَهْرٌ, هَاجِرَةٌ, sedangkan waktu sore terbagi menjadi 5 keadaan: خَدُورٌ, غُرُوبٌ, صُبُوبٌ, أُصَيْلٌ, عَصْرٌ.

#### 2.1.2 Ilmu Tashrif dalam Struktur Bahasa Arab

Dalam struktur bahasa Arab, dikenal 2 disiplin ilmu utama, yaitu : *nahwu* dan *shorof/ tashrif*. Nahwu adalah metode analisis jenis kata dalam suatu kalimat. Dengan nahwu, kedudukan / jabatan kata dalam kalimat dapat diidentifikasi. Sedangkan ilmu *tashrif* adalah ilmu yang mempelajari kemungkinan perubahan suatu kata sebelum kata tersebut masuk dalam suatu kalimat.

Ilmu *tashrif* mempelajari perubahan bentuk dan pola suatu kata. Teks kamus bahasa Arab seringkali menempatkan susunan suatu kata dalam urutan

Kata dasar dalam kamus tersebut biasanya adalah kata kerja bentuk lampau (*fi'il madhi*) atau kata benda bentukan dari kata kerja (*isim masdar*).

Tabel 2.5 Kata Benda Penunjuk Waktu dengan Perasaan Kebahasaan yang Detail

Kata	Makna
ذُرُورٌ	Waktu mula-mula timbul matahari di waktu pagi
بُزُوعٌ	Waktu mula timbul matahari selepas waktu <i>dazuur</i>
ضُحَى	waktu matahari naik setinggi tombak
غَزَالَةٌ	waktu matahari naik setelah dhuha
هَاجِرَةٌ	waktu tengah hari yang mulai terasa panas
ظُهُرٌ	waktu tengah hari matahari mulai naik menegak
زَوَالٌ	waktu matahari berada tegak di atas kepala
عَصْرٌ	waktu siang mulai berakhir, matahari kemerah-merahan
أَصِيلٌ	waktu matahari mulai condong ke arah barat
صُبُوبٌ	waktu matahari semakin menghilang
غُرُوبٌ	Waktu matahari mula terbenam
خَدُورٌ	waktu matahari hilang dari pandangan atau gelap

Ilmu *tashrif* mempelajari perubahan bentuk dan pola suatu kata. Teks kamus bahasa Arab seringkali menempatkan susunan suatu kata dalam urutan berdasarkan kata dasarnya. Kata dasar dalam kamus tersebut biasanya adalah kata kerja bentuk lampau (*fi'il madhi*) atau kata benda bentukan dari kata kerja (*isim mashdar*). Sebagai contoh, untuk mencari kata *يَغْسِلُ* dalam kamus, seseorang tidak bisa mencarinya dalam deretan kata berawalan *ي* (huruf *ya'*), karena susunan

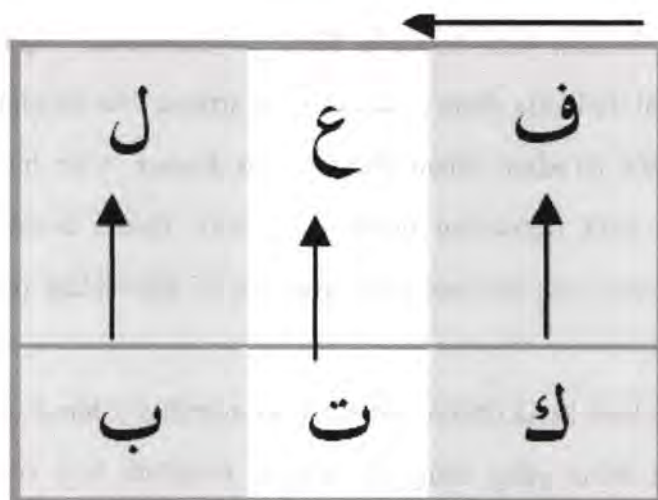
kata diurutkan menurut kata dasarnya. Kata *يَغْسِلُ* bermakna ' (dia laki-laki) sedang mencuci' berasal dari kata dasar : *غَسَلَ* yang artinya: 'dia laki-laki telah mencuci'. Karena itu, kata tersebut dalam teks tertulis kamus Arab biasanya diletakkan dalam deretan kata berawalan huruf *غ* (ghain), bukan huruf *ي* (ya'). Untuk menganalisis suatu kata menjadi kata dasarnya ini diperlukan pengetahuan dalam ilmu *tashrif*.

Mayoritas kata kerja dalam bahasa Arab memiliki 3 huruf utama. Karena itu, 3 huruf utama inilah yang dianggap sebagai *template* bagi susunan huruf kata kerja yang lain. Template susunan huruf tersebut adalah *فَعَلَ*. Merujuk pada *template* ini, huruf inti pertama dianggap sebagai *ف*, huruf inti ke-2 dianggap sebagai huruf *ع*, dan huruf inti ke-3 dianggap sebagai *ل*. Sebagai contoh, kata: *كَتَبَ* yang artinya : 'telah menulis' merupakan morfologi turunan dari *template فَعَلَ* (Muhammad, MI, 2006).

Gambar 2.1 menjelaskan pensejajaran masing-masing huruf pembentuk kata dengan huruf pada templatnya. Huruf *ك* adalah huruf pertama, sehingga merupakan turunan dari huruf *ف* pada *template*. Huruf *ت* adalah huruf kedua dalam susunan huruf, merupakan representasi dari huruf *template ع*. Sedangkan huruf terakhir *ب* merupakan pasangan dari huruf *template ل*. *Template* huruf penyusun tersebut dalam istilah *tashrif* disebut sebagai *wazan*, yang bermakna timbangan. Artinya, semua huruf penyusun suatu kata akan direferensikan pada *wazan/* timbangannya yang sesuai.

Jumlah *wazan* sebagai referensi dalam *tashrif* ada 35 buah (Maksum bin Ali, 1965). Kata yang dijadikan sebagai acuan dalam *wazan* adalah kata kerja bentuk lampau (*fi'il madhi*) dan kata kerja sekarang/ akan datang (*fi'il mudhari*). Tabel 2.6 menunjukkan daftar *wazan* yang dijadikan acuan dalam *tashrif system* yang akan dikembangkan.

Seluruh kata kerja akan terklasifikasikan dalam *wazan* masing-masing. Misalnya, kata *نَصَرَ* (menolong) memiliki *wazan فَعَّلُ - يَفْعُلُ* (no 1). Contoh



Gambar 2.1 Template Huruf Pembentuk Kata Kerja

kata kerja lain yang termasuk wazan ini adalah قَعَدَ (duduk) dan كَتَبَ (telah menulis). *Wazan* يَفْعُلُ - فَعَلَ adalah referensi dari kata kerja semisal ضَرَبَ (memukul) dan جَلَسَ (duduk). Teks kamus bahasa Arab yang ada akan menuntun seseorang untuk mengetahui pada *wazan* apa suatu kata bisa diklasifikasikan.

Dua jenis kata yang dijadikan acuan sebagai *wazan* adalah kata kerja lampau (*fi'il madhi*) dan kata kerja masa sekarang atau akan datang (*fi'il mudhari'*). Masih ada jenis kata lain yang menjadi obyek kajian ilmu *tashrif* dalam memetakan morfologi suatu kata. Tabel 2.7 menjelaskan jenis-jenis kata yang ada dalam ilmu *tashrif*.

*Fi'il madhi* adalah kata kerja lampau yang menunjukkan bahwa kata kerja itu telah dilakukan. Kata kerja yang menunjukkan perbuatan pada saat sekarang (sedang berlangsung) atau akan datang adalah *fi'il mudhari'*. Konteks kalimatlah yang akan menentukan makna kata tersebut berada pada masa saat ini atau akan datang. *Mashdar* adalah kata benda yang diturunkan dari kata kerjanya. Sebagai contoh, dalam bahasa Indonesia, kata 'tulisan' adalah kata benda yang diambil dari kata kerja 'menulis'. Demikian juga kata 'pemberian' adalah kata benda yang diturunkan dari kata kerja 'memberi'. *Mashdar* dalam bahasa Arab dibedakan menjadi 2, yaitu *mashdar biasa* dan *mashdar miim*, yaitu bentuk *mashdar* yang didahului oleh huruf م (*mim*). *Fa-'il* adalah kata benda pelaku, atau

Tabel 2.6 Daftar *Wazan*

NO	WAZAN	NO	WAZAN	NO	WAZAN
1	فَعَلَ - يَفْعُلُ	13	فَعْنَلْ - يُفْعِنُلُ	25	اِفْعَوَلْ - يَفْعَوُلُ
2	فَعِلْ - يَفْعِلُ	14	فَعْلٌ - يُفْعَلُ	26	تَفْعَلَلْ - يَتَفْعَلَلُ
3	فَعَلْ - يَفْعَلُ	15	فَاعَلْ - يُفَاعِلُ	27	تَفْوَعَلْ - يَتَفْوَعَلُ
4	فَعِلْ - يَفْعِلُ	16	أَفْعَلْ - يُفْعَلُ	28	تَمْفَعَلْ - يَتَمْفَعَلُ
5	فَعْلٌ - يَفْعَلُ	17	تَفَاعَلْ - يَتَفَاعَلُ	29	تَفْيَعَلْ - يَتَفْيَعَلُ
6	فَعِلْ - يَفْعِلُ	18	تَفْعَلْ - يَتَفْعَلُ	30	تَفْعَوَلْ - يَتَفْعَوُلُ
7	فَعْلَلْ - يُفْعَلَلُ	19	اِفْتَعَلْ - يَفْتَعِلُ	31	تَفْيَعِلْ - يَتَفْيَعِلُ
8	فَوَعَلْ - يُفْوَعِلُ	20	اِنْفَعَلْ - يَنْفَعِلُ	32	تَفْعَلَى - يَتَفْعَلَى
9	فَيْعَلْ - يُفَيْعَلُ	21	اِفْعَلْ - يَفْعَلُ	33	اِفْعَنْلَلْ - يَفْعَنْلَلُ
10	فَعَوَلْ - يُفْعَوُلُ	22	اِسْتَفْعَلْ - يَسْتَفْعِلُ	34	اِفْعَنْلَى - يَفْعَنْلَى
11	فَعِيلْ - يُفَعِيلُ	23	اِفْعَوَعَلْ - يَفْعَوَعِلُ	35	اِفْعَلَلْ - يَفْعَلَلُ
12	فَعْلَى - يُفَعْلَى	24	اِفْعَالٌ - يَفْعَالُ		

dalam bahasa Indonesia dikenal sebagai subyek. Kata benda yang menunjukkan obyek penderita disebut sebagai *maf'ul*.

Untuk mengekspresikan bentuk perintah, bisa digunakan *fi'il amr*. *Fi'il amr* merupakan kata kerja perintah untuk melakukan suatu hal, yang merupakan kebalikan dari *fi'il naahi* yaitu larangan untuk melakukan suatu hal. *Isim zamaan* adalah kata benda yang menunjukkan keterangan waktu dan memiliki morfologi yang sama persis dengan *isim makaan* (kata benda yang menunjukkan keterangan tempat). Untuk membedakan keduanya, bisa dilihat pada konteks kalimat yang digunakan.

Tabel 2.7 Jenis Kata dalam *Tashrif*

No	Tashrif	Indonesia	Keterangan
1	فعل ماض	<i>Fi'il Madhi</i>	Kata kerja lampau
2	فعل مضارع	<i>Fi'il Mudhari'</i>	Kata kerja sekarang/akan datang
3	مصدر	<i>Mashdar</i>	Kata benda dari kata kerja
4	مصدر ميم	<i>Mashdar Mim</i>	Mashdar yang diawali huruf 'mim'
5	فاعل	<i>Fa'il</i>	Kata benda pelaku
6	مفعول	<i>Maf'ul</i>	Kata benda obyek
7	فعل أمر	<i>Fi'il Amr</i>	Kata kerja perintah
8	فعل ناهي	<i>Fi'il Nahi</i>	Kata kerja larangan
9	إسم زمان و مكان	<i>Isim Zamaan/Makaan</i>	Kata benda waktu dan tempat
10	إسم الألة	<i>Isim Alat</i>	Kata benda alat

*Isim alat* adalah kata benda yang menunjukkan alat / *tools* yang mendukung terlaksananya kata kerja tertentu. Sebagai contoh, jika kata kerjanya adalah فَتَحَ yang berarti 'telah membuka', maka *isim alatnya* adalah مِفْتَاحٌ yang berarti 'alat untuk membuka', yang sering juga diterjemahkan sebagai 'kunci'.

Ilmu *tashrif* mempelajari template pada masing-masing jenis kata berdasarkan *wazan* yang sesuai. Literatur – literatur ilmu *tashrif* yang ada memberikan tuntunan template pada masing-masing jenis kata sesuai dengan *wazannya*. Berdasarkan template tersebut, setelah diketahui *fi'il madhi* suatu kata dengan *wazannya*, maka jenis kata yang lain akan bisa diprediksi.

Kata كَتَبَ yang berarti 'telah menulis' adalah *fi'il madhi* (kata kerja lampau). Kata ini masuk dalam *wazan* فَعَلَ-يَفْعُلُ. Berdasarkan template yang ada, kata kerja bentuk sekarang/ akan datang (*fi'il mudhari'*)nya bisa ditentukan, yaitu:

يَكْتُبُ (sedang/akan menulis). *Mashdar* dan *mashdar mim*-nya adalah كَتَبَ dan مَكْتُبٌ. Pelaku/ *fa'il* dari kata kerja ini adalah كَاتِبٌ yang berarti 'penulis'. Sedangkan *maf'ulnya* adalah مَكْتُوبٌ yang berarti 'sesuatu yang ditulis'. Bentuk perintah atau *fi'il amrnya* adalah اُكْتُبْ yang bermakna 'tulislah'. Untuk menyatakan larangan: 'jangan menulis', kita bisa menyatakan dengan *fi'il nahi*, yaitu: لَا تَكْتُبْ. *Isim zamaan* dan *makaan* untuk kata kerja كَتَبَ adalah مَكْتُبٌ. *Isim alat* untuk kata kerja كَتَبَ adalah مِكْتَبٌ yang berarti 'alat untuk menulis'.

Contoh penerapan kata كَتَبَ pada masing-masing jenis kata sesuai *wazamya* bisa dilihat pada Tabel 2.8. Proses analisis morfologi kata (*tashrif*) pada tiap-tiap jenis kata ini disebut sebagai *tashrif istilahy*. Sedangkan proses *tashrif* suatu kata pada jenis kata tertentu berdasarkan kata ganti personalnya disebut sebagai *tashrif lughowy*. Jumlah kata ganti personal yang dikenal dalam *tashrif* ada 14 (Maksum bin Ali, 1967). Kata ganti personal tersebut dinamakan *dhomir*. Pada sebagian literatur jumlah *dhomir* tidak sampai berjumlah 14, karena ada beberapa *dhomir* yang sama untuk beberapa kata ganti. Penjelasan tentang masing-masing *dhomir* tersebut ditunjukkan pada Tabel 2.9.

*Dhomir* هُوَ bisa diartikan dalam bahasa Indonesia sebagai kata ganti 'dia laki-laki'. Pada tabel 2.9 dijelaskan keterangan pada masing-masing jenis *dhomir*. Keterangan tersebut meliputi keadaan person yang dibicarakan (orang ke berapa), jumlah person, dan jenis personnya (laki-laki atau perempuan). Sebenarnya kata ganti ini tidak hanya berlaku bagi manusia, karena dalam bahasa Arab benda-benda yang tidak berakal juga diklasifikasikan menjadi jenis laki-laki dan perempuan. Orang ke-1 berarti orang yang berbicara, orang ke-2 adalah orang yang diajak bicara, sedangkan orang ke-3 adalah figur yang sedang dibicarakan. Dalam *tashrif lughowy*, analisis morfologi kata dilakukan terhadap suatu jenis kata untuk masing-masing *dhomirnya*. Untuk *fi'il madhi*, ditashrif berdasarkan semua *dhomir*. Demikian juga pada *fi'il mudhari'* dan seluruh jenis kata yang ada. Tabel 2.10 menjelaskan *tashrif lughowy* untuk *fi'il madhi* كَتَبَ.



Tabel 2.8 Contoh penempatan kata كَتَبَ pada masing-masing jenis kata

Jenis Kata	Kata Arab	Arti
<i>Fi'il Madhi</i>	كَتَبَ	telah menulis
<i>Fi'il Mudhari'</i>	يَكْتُبُ	sedang/ akan menulis
<i>Mashdar</i>	كُتِبَ	Tulisan
<i>Mashdar mim</i>	مَكْتُبٌ	Tulisan
<i>Fa'il</i>	كَاتِبٌ	Penulis
<i>Maf'ul</i>	مَكْتُوبٌ	yang ditulis
<i>Fi'il Amr</i>	اُكْتُبْ	Tulislah
<i>Fi'il Nahi</i>	لَا تَكْتُبْ	jangan tulis
<i>Isim Zamaan/ Makaan</i>	مَكْتَبٌ	tempat/ waktu menulis
<i>Isim Alat</i>	مِكْتَبٌ	alat untuk menulis

*Dhomir* pada Tabel 2.10 memiliki nomor yang sama dengan Tabel 2.9. Kata كَتَبَ memiliki *dhomir* هُوَ yang berarti 'dia laki-laki telah menulis'. *Fi'il madhi* untuk kata ganti orang ke-2 berjumlah 2 orang sama untuk jenis laki-laki dan perempuan. Hal ini bisa dilihat pada Tabel 2.10 nomor 8 dan nomor 11. Kata كَتَبْنَا bisa menunjukkan kata ganti 'kita' atau 'kami', tergantung konteks makna dalam kalimat.

Berdasarkan jumlah huruf pembentuknya, suatu *wazan* diklasifikasikan dalam 4 jenis (Maksum bin Ali, 1967) seperti dalam Tabel 2.11.

Jenis الثلاثي المحرد menunjukkan bahwa suatu kata terdiri dari 3 huruf asli. *Wazan* yang masuk kategori ini di antaranya adalah nomor 1-6 pada Tabel 2.6. Sedangkan pada الثلاثي المزيد tiga huruf pembentuk asli ditambah huruf tambahan. Sebagai contoh, *wazan* فَعَّلَ ada penambahan huruf ع yang dibaca *tasydid* sehingga terhitung dua. *Wazan* فَاعَلَ ada tambahan huruf alif (ا), أَفَعَلَ mengalami tambahan

huruf *أ* di awal kata, *تَفَاعَلَ* mengalami tambahan huruf *ت* di awal, dan huruf alif di tengah.

Tabel 2.9 *Dhomir* (kata ganti personal) dalam *tashrif*

No	Dhamir	Keterangan
1	هُوَ	Orang ke-3, tunggal, laki-laki
2	هُمَا	Orang ke-3, dua, laki-laki
3	هُمْ	Orang ke-3, lebih dari dua, laki-laki
4	هِيَ	Orang ke-3, tunggal, perempuan
5	هُمَا	Orang ke-3, dua, perempuan
6	هُنَّ	Orang ke-3, lebih dari dua, perempuan
7	أَنْتَ	Orang ke-2, satu orang, laki-laki
8	أَنْتُمَا	Orang ke-2, dua, laki-laki
9	أَنْتُمْ	Orang ke-2, lebih dari dua, laki-laki
10	أَنْتِ	Orang ke-2, tunggal, perempuan
11	أَنْتُمَا	Orang ke-2, dua, perempuan
12	أَنْتُنَّ	Orang ke-2, lebih dari dua, perempuan
13	أَنَا	Orang ke-1, tunggal, laki-laki/ perempuan
14	نَحْنُ	Orang ke-1, lebih dari satu, laki-laki/perempuan

Jika ditinjau dari segi bangunan kata (*bina'*), *tashrif* mengenal kata-kata yang tidak mengandung huruf *illat* (ي, و, dan ا) dan kata yang mengandung huruf *illat* tersebut. Adanya huruf *illat* tersebut memerlukan perlakuan khusus sehingga memiliki template yang berbeda. Macam-macam bangunan kata yang dikenal dalam *tashrif* ada 13 (Maksum bin Ali, 1967), sebagaimana ditunjukkan pada Tabel 2.12.

Tabel 2.10 *Tashrif Lughowy* untuk *fi'il madhi* كَبَّ

No	Fi'il Madhi	Arti
1	كَتَبَ	Dia laki-laki telah menulis
2	كَتَبَا	Mereka 2 orang laki-laki telah menulis
3	كَتَبُوا	Mereka lebih dari 2 laki-laki telah menulis
4	كَتَبَتْ	Dia satu perempuan telah menulis
5	كَتَبَتَا	Mereka 2 perempuan telah menulis
6	كَتَبْنَ	Mereka lebih dari 2 perempuan telah menulis
7	كَتَبْتَ	Engkau 1 laki-laki telah menulis
8	كَتَبْتُمَا	Engkau 2 laki-laki telah menulis
9	كَتَبْتُمْ	Kalian lebih dari 2 laki-laki telah menulis
10	كَتَبْتِ	Engkau 1 perempuan telah menulis
11	كَتَبْتُمَا	Kalian 2 perempuan telah menulis
12	كَتَبْتُنَّ	Kalian lebih dari 2 perempuan telah menulis
13	كَتَبْتُ	Aku telah menulis
14	كَتَبْنَا	Kami/ kita telah menulis

Tabel 2.11 Jumlah Huruf Pembentuk Kata

No	Jml Huruf	Keterangan
1	الثلاثي المجرد	3 huruf asli
2	الثلاثي المزيد	3 huruf dengan tambahan
3	الرباعي المجرد	4 huruf asli
4	الرباعي المزيد	4 huruf dengan tambahan

Bangunan kata *shahih* menunjukkan bahwa kata tersebut tidak memiliki huruf *illat* sebagai unsur penyusunnya. Contoh kata jenis ini adalah كَتَبَ (telah menulis), جَلَسَ (telah duduk), dan نَصَرَ (telah menolong). Bangunan kata *mudha'af* artinya ada huruf yang *ditasydid*. Kata-kata yang termasuk jenis ini di antaranya كَبَّرَ (bertakbir) dan قَرَّبَ (mendekatkan). *Mahmuuz* artinya bangunan suatu kata yang mengandung huruf ا. Jika letaknya di awal, disebut *mahmuuz fa'* karena mensubstitusi *fa' fi'il*, sedangkan jika di huruf ke-2 disebut *mahmuuz 'ain*, dan jika pada huruf ke-3 disebut *mahmuuz lam*.

*Mitsaal* adalah bangunan kata yang huruf pertamanya adalah ي (ya') atau و (wawu). Jika huruf pertama adalah ya' maka disebut sebagai *mitsaal yaa-i*, sedangkan apabila huruf pertama adalah wawu, maka disebut sebagai *mitsaal wawi*. Bangunan kata *ajwaf* artinya di tengah-tengah kata tersebut terdapat tanda huruf ya' atau wawu.

Jika yang direpresentasikan adalah huruf ya' maka disebut *ajwaf yaa-i*, namun jika huruf wawu, disebut sebagai *ajwaf waawi*. Contoh *ajwaf yaa-i* adalah kata تَبَيَّنَ (telah menjelaskan), sedangkan pada *ajwaf wawi* contohnya adalah طَالَ (telah memanjang). Kata طَالَ sepintas tidak mengandung huruf wawu, padahal sebenarnya tanda baca ا di tengah tersebut disubstitusi oleh huruf wawu. Untuk mengetahui substitusi tersebut bisa dilihat pada *fi'il mudhari'nya* yang merupakan kata يَطْوُلُ (sedang memanjang).

Jika huruf *illat ya'* atau *wawu* terletak di akhir suatu kata, bangunan katanya disebut sebagai *Naqish*. Jika huruf *illat* itu adalah ya' maka disebut sebagai *naqish yaa-i*. Sebaliknya jika huruf *illatnya* adalah wawu, maka disebut sebagai *naqish wawi*. Contoh *naqish ya-i* adalah رَضِيَ (telah ridla). Kata غَزَا adalah sebuah permisalan dari bangunan kata *naqish wawi*.

Tabel 2.12 Macam-macam Bangunan Kata (*Bina'*) dalam *Tashrif*

No	Jenis	Indonesia	Keterangan
1	صحيح	<i>Shahih</i>	tidak mengandung huruf 'illat
2	مضاعف	<i>Mudha'af</i>	2 huruf berdekatan sama dan di- <i>tasydiid</i>
3	مهموز فاء	<i>Mahmuz Fa'</i>	Huruf pertama adalah hamzah
4	مهموز عين	<i>Mahmuz 'Ain</i>	Huruf asli kedua adalah hamzah
5	مهموز لام	<i>Mahmuz Laam</i>	Huruf asli ketiga adalah Laam
6	مثال واوي	<i>Mitsaal wawi</i>	Huruf asli pertama adalah wawu
7	مثال يائي	<i>Mitsaal Yaa-i</i>	Huruf asli pertama adalah yaa'
8	أجوف واوي	<i>Ajwaf waawi</i>	Huruf asli pertengahan (kedua) adalah wawu
9	أجوف يائي	<i>Ajwaf Yaa-i</i>	Huruf asli pertengahan (kedua) adalah yaa'
10	ناقص واوي	<i>Naqish waawi</i>	Huruf asli akhir adalah wawu
11	ناقص يائي	<i>Naaqish Yaa-i</i>	Huruf asli akhir adalah yaa'
12	لفيف مقرون	<i>Lafiif maqruun</i>	Mengandung 2 huruf 'illat yang bersebelahan
13	لفيف مفروق	<i>Lafiif mafruuq</i>	Mengandung 2 huruf 'illat yang tdk bersebelahan

Contoh peletakan *template* pada literatur referensi (Maksum bin Ali, 1967) bisa dilihat pada Tabel 2.13.

Peletakan *template* di Tabel 2.13 adalah untuk jumlah huruf الثلاثي المحرد dan wazan فَعَلَ-يَفْعُلُ. Template – template tersebut diletakkan dengan orientasi pada contoh kata karena sasaran pengguna template ini adalah manusia (untuk memudahkan hafalan).

Tabel 2.13 Contoh Peletakan Template pada Literatur *al-Amtsilatut Tashrifiyah*

JUMLAH HURUF :      النّالائي ائخرود      , 3 HURUF ASLI.										
WAZAN :      فعّل - فعّعل										
فعل ماض	فعل مضارع	مصدر ميم	فاعل	مفعول	فعل امر	فعل ناهي	اسم زمان و مكان	اسم الألة		
فعّل	فعّعل	فعّلا	فاعّل	مفعّول	أفّعّل	لا فعّعل	مفعّعل	مفعّعل	صحیح	
معدّ	معدّلا	معدّلا	معدّ	معدّول	معدّ	لا معدّ	معدّ	معدّ	مضاعف	
صان	صوّن	صونا	صانّ	مصوّن	صنّ	لا صنّ	مصانّ	مصوّن	أجوف واري	
عرا	عروّ	عروّا	عار	معروّ	أعروّ	لا أعروّ	معروّ	معروّ	ناقص واري	
أصّ	أصّعل	أصّلا	أصّ	أصّول	أصّعل	لا أصّعل	أصّعل	أصّعل	مهموز فاء	

## 2.2 Penjelasan tentang Machine Translation

Pendekatan *example-based* dalam bidang *Machine Translation (MT)* saat ini menjadi semakin populer. Menentukan definisi yang tepat untuk *Example-Based Machine Translation (EBMT)* adalah suatu hal yang sulit dilakukan. Meskipun definisi dari system MT dikenal kompleks, tetap dilakukan upaya untuk mendefinisikan EBMT sebagai pendekatan yang berbeda dibandingkan arsitektur MT yang lain (RBMT dan SMT).

Kerangka kerja (*framework*) yang dominan sampai akhir 1980-an adalah apa yang saat ini dikenal sebagai *Rule-Based Machine Translation (RBMT)*. Sejak saat itu, riset didominasi oleh pendekatan *corpus-based*, yang berbeda dengan *Statistical Machine Translation (SMT)* yang berdasarkan frekuensi kata dan kombinasi kata. Sedangkan EBMT berdasarkan ekstraksi dan kombinasi frase (atau bagian pendek lain dari sebuah teks).

SMT –pertama kali diformulasikan pada tahun 1988 oleh IBM- konsepnya adalah : Kalimat-kalimat pada bilingual corpus pertama – tama disejajarkan (*aligned*), selanjutnya kata-kata dari SL (*source language*) dan TL (*target language*) disejajarkan secara bersesuaian. Berdasarkan pensejajaran ini diturunkanlah model terjemahan (*translation model*) dari frekuensi SL-TL dan

sebuah model bahasa (*language model*) dari urutan kata TL. Terjemahan meliputi **pemilihan kata-kata TL yang paling sesuai untuk masing-masing input kata dan penentuan urutan yang paling tepat dari kata-kata yang terpilih dalam TL**. Unit dasar dari sistem SMT adalah kata, namun saat ini segment yang lebih luas juga diperhatikan.

Mendefinisikan model EBMT tidaklah semudah pada SMT. Secara dasar – meskipun agak dangkal- sebuah sistem adalah sistem EBMT jika ia menggunakan segmen-segmen (urutan kata / string dan bukan kumpulan kata secara individu) dari bahasa sumber (SL) yang diekstraksi dari sebuah corpus teks (database contoh) untuk membangun teks-teks pada sebuah bahasa target dengan arti yang sama. Unit dasar dari EBMT adalah urutan kata-kata tersebut (frase).

EBMT adalah ramuan dari berbagai metode dan teknik yang sebagian diambilkan dari RBMT, sebagian lagi dari SMT, atau *Translation Memory* (TM), dsb. Secara khusus, tidak ada konsesus tentang apakah EBMT itu atau apa yang tidak bisa disebut sebagai EBMT. Pada pengenalan kumpulan paper tentang EBMT, sang editor menahan diri dari memberikan definisi, beralasan bahwa bidang sains bisa berhasil baik tanpa kerangka kerja yang kedap, tentu saja akan tumbuh dengan subur karena tidak terlalu didefinisikan.

### **Konsepsi Orisinal dari EBMT**

Sebagaimana definisi pendahuluan di atas, kita bisa mengidentifikasi proses dasar dari EBMT sebagai : pensejajaran teks, menepatkan (*matching*) input kalimat dengan frase (contoh) pada corpus, pemilihan dan ekstraksi dari frase TL yang ekuivalen, dan adaptasi serta mengkombinasikan frase TL sebagai kalimat output yang bisa diterima.

Dalam konsepsi orisinalnya ( Nagao 1984), EBMT kelihatannya dipertimbangkan sebagai media untuk mengatasi kekurangan dari sistem RBMT, mengidentifikasi kelemahannya ketika menterjemahkan bahasa-bahasa yang sangat berbeda strukturnya, seperti Bahasa Inggris dengan Jepang, dan juga diharapkan bisa menghasilkan output dengan kualitas baik – khususnya dalam menangani *collocation* (susunan kata) , contohnya terjemahan *yabureru* pada : *The bag was broken* dan *The President was defeated in the election*.

Contoh lain adalah perbedaan terjemahan kata Jepang *kakeru* berdasarkan konteks:

*Hang* something on a tree, *put* something on / over one's shoulder, *cover* someone or something. Contoh-contoh yang ada selanjutnya akan diperlakukan seperti data SL dan TL lain, yaitu dalam representasi pohon (*tree*). Karena itu, input kalimat akan dianalisis sebisa mungkin, dan dilakukan *transfer* menggunakan contoh-contoh yang telah diinisiasi ketika aturan-aturan dan *trees* gagal.

Beberapa peneliti (diantaranya Sumita, 1990) menggunakan contoh-contoh untuk meningkatkan system RBMT dan tidak yakin EBMT berkontribusi pada seluruh proses penerjemahan, sedangkan peneliti yang lain (Sato dan Nagao, 1990) terdorong untuk meneliti system EBMT murni, dimana proses dasar dibentuk dengan mencari contoh dari kalimat TL yang analog dengan kalimat input SL, dan **aturan baru diterapkan hanya jika contoh-contoh tidak bisa ditemukan dalam database.**

Di satu sisi, metode example-based digunakan dalam ruang yang secara dasar system RBMT diterapkan dan secara esensial terlihat sebagai pengembangan dari tradisi MT, dan di satu sisi terdapat suatu ketetapan bahwa EBMT merepresentasikan paradigma baru ( secara personal, penulis segan untuk menyebut paradigma baru, ia lebih senang dengan istilah arsitektur atau kerangka kerja).

Satu alasan dalam mengeksplorasi pendekatan EBMT adalah : karena ia berdasar teks aktual, maka output terjemahan seharusnya lebih mudah terbaca dan lebih sensitive dalam konteks dibandingkan system RBMT, sehingga memiliki kualitas yang lebih tinggi dalam hal kesesuaian dan penanganan masalah 'ungkapan'(idiom). Alasan yang kedua adalah bahwa EBMT lebih mudah dikembangkan dengan menambahkan contoh-contoh dari *bilingual corpora* , sedangkan pengembangan system RBMT meliputi modifikasi dan penambahan aturan-aturan kompleks dan *lexical entries*. Alasan ketiga adalah bahwa EBMT tidak melibatkan kompleksitas leksikal dan *structural transfer* yang ditemukan pada kebanyakan sistem RBMT saat ini, sehingga arsitektur dasar EBMT lebih sederhana dan lebih sedikit kecenderungan errornya dibandingkan RBMT. Alasan ke-empat, EBMT bisa menangani kasus-kasus terjemahan yang melibatkan



bahasa dengan perbedaan struktur yang sangat tinggi dan pilihan lexical yang halus, yang RBMT sering gagal dalam hal itu. Secara umum, alasan utama (pemilihan) EBMT adalah potensinya untuk meningkatkan pembangkitan kalimat terjemahan (TL).

#### **Definisi EBMT oleh Somers, Turcato, dan Popowich**

Harold Somers (1999) mengemukakan tinjauan singkat secara garis besar tentang EBMT. Pemaparannya ini meliputi : isi, ukuran, dan organisasi database dari parallel bilingual text corpora – bagaimana teks tersebut dipilih dan diedit (untuk mengurangi redundansi dan contoh tidak umum yang bisa mengacaukan), bagaimana pensejajarannya (*aligned*), proses taggingnya, dianalisis sebagai representasi *tree*, dan sebagainya. Demikian juga, terdapat pilihan dalam proses *matching* (berdasarkan character, kata, struktur), pengukuran kemiripan (*similarity*) – misalnya berdasarkan statistik atau berdasarkan referensi *thesauri*, adaptasi dari contoh yang diekstraksi, dan rekombinasi (*recombination*) yang dilakukan untuk menghasilkan kalimat TL. Dia menunjukkan bahwa *recombination*, meskipun memegang peranan penting dalam EBMT (yang tujuan utamanya adalah menghasilkan output berkualitas lebih baik dari RBMT), tetapi seringkali diabaikan dalam riset. Akhirnya, Somers menyebutkan garis besar aplikasi EBMT secara aktual maupun potensi yang dimilikinya dibandingkan arsitektur MT yang lain, khususnya penurunan kamus dan aturan *grammar* dari sistem RBMT, dan peranan EBMT dalam *multi-engine* dan sistem hybrid.

Somers secara tepat menunjukkan bahwa penggunaan dari apa yang diklaim sebagai metode EBMT tidak serta merta berarti bahwa sistem itu adalah EBMT. Variasi metode dan teknik, dalam cara bagaimana ia berinteraksi, adalah seluruh indikator dari kerangka kerja riset yang produktif dan terus berkembang, tapi hal itu tidak membuat pendefinisian menjadi semakin mudah. Apa yang dilihat Somers sebagai suatu intisarinya? Pertama, penggunaan sebuah *bilingual corpus* adalah merupakan bagian dari definisi, tapi hal itu saja tidaklah cukup, karena hampir seluruh riset MT saat ini (termasuk sistem RBMT) menggunakan *teks corpora* untuk mendefinisikan dan membatasi rentang data yang akan *discover* – paling tidak pada tahapan awal pengembangan.

Sebagai definisi yang lebih dekat, Somers mengajukan : " EBMT berarti basis pengetahuan utama berasal dari contoh". Tetapi, kalimat contoh bisa saja digunakan dalam sistem RBMT sebagai sumber data dimana aturan-aturan yang digeneralisasi dan pola-pola diturunkan darinya. Demikian juga, database dari sistem SMT diturunkan dari corpora teks contoh. Karakteristik yang lebih khusus dari EBMT sebenarnya adalah : "contoh-contoh yang ada digunakan pada *run-time*". Berdasarkan komentar dari Somers, definisi ini akan mengeluarkan SMT dari kerangka kerja EBMT, karena data yang digunakan pada SMT diturunkan pada pengembangan dari proses translasi. Sebagai tambahan, kondisi '*run-time*' terlihat mengeluarkan banyak sistem EBMT yang dideskripsikan dalam koleksi *Carl-Way*.

Dalam sebuah artikel yang mengikuti ikhtisar yang dibuat Somer, Davide Turcato dan Fred Popowich mengambil pokok persoalan pada definisi Somer. Tujuan mereka adalah membuat sebuah kerangka kerja untuk mendefinisikan proses utama dari EBMT, yaitu untuk mengidentifikasi atau mengisolasi apa yang membuat sebuah sistem sebagai *example-based* , dan bukan sebagai *rule-based*. Pertama, mereka setuju bahwa penggunaan sebuah database contoh semata dalam sebuah sistem MT tidaklah kemudian berarti sistem itu EBMT, karena (mereka beralasan) cara-cara tentang bagaimana pengetahuan didapatkan atau diekspresikan bukanlah suatu hal yang relevan (untuk dibahas dan dibedakan), namun pokok permasalahannya adalah bagaimana pengetahuan itu digunakan dalam operasi. Pada dasar ini mereka membandingkan prinsip-linguistik sistem EBMT dan satu tipe dari sistem RBMT berbasis transfer (lexicalist 'shake dan bake') – karena tipe ini (tidak seperti sistem RBMT yang lain) juga menghindari transfer struktural. Tujuannya adalah untuk memperjelas status dari database contoh.

Jika bisa ditunjukkan bahwa operasi EBMT ekuivalen dengan sebuah sistem yang tidak menggunakan database contoh, maka EBMT harus didefinisikan dalam keadaan tidak memiliki referensi ke database contoh, atau karakteristik lain EBMT ditinjau dari *knowledge acquisition*, bukan dalam segi penggunaan pengetahuan – yang dalam tinjauan Somer, hal-hal ini telah ditolak sebagai sebuah dalam mendefinisikan karakteristik yang valid. Pertanyaan penting

adalah : " Bagaimana kalimat-kalimat dipecah selama proses *matching* EBMT kalau dibandingkan dengan proses *decomposition* (analisis) pada lexicalist RBMT". Beberapa sistem MT harus menangani konstruksi-konstruksi yang tidak bisa diterjemahkan berdasarkan komposisi; tapi ia membutuhkan akses ke tempat penyimpanan (repository) '*non-monotonic context*' (contoh).

Pada RBMT, tempat penyimpanan diekstraksi (dibuat) dari kamus atau teks sumber; sedangkan pada EBMT tempat penyimpanan yang digunakan dalam operasi bisa juga diekstraksi dari sumber (database contoh) sebagai '*explicit knowledge*'. Pada kasus ini, informasi linguistik yang digunakan oleh EBMT tidak bisa dipisahkan dari informasi yang digunakan oleh lexicalist MT. Bagaimanapun, pada arsitektur EBMT yang lain, bisa terdapat referensi langsung ke database contoh selama pemrosesan kalimat-kalimat (selama penerjemahan). Pada kasus ini tempat penyimpanan digunakan sebagai database '*implicit knowledge*'. Turcato dan Popowich beralasan bahwa keadaan ini hanya bisa dilakukan jika EBMT memiliki akses dan menggunakan database contoh orisinal secara penuh selama proses terjemahan, yang membedakan EBMT dengan RBMT. Dengan kata lain, konsepsi orisinal tentang terjemahan dengan analogi (yang pertama kali diperkenalkan oleh Nagao) merepresentasikan teknik paling spesifik untuk EBMT dan itu menunjukkan bahwa EBMT adalah satu-satunya sistem yang dirangsang untuk memanfaatkan seluruh contoh yang ada. Akses penuh semacam itu hanya tersedia jika sistem EBMT masih belum memproses contoh – contoh (sebagai '*explicit knowledge*').

Dengan kata lain, mereka menyarankan bahwa sistem yang benar-benar murni EBMT adalah ketika informasi tidak mengalami *pre-processing* tersedia utuh dan belum dianalisis melalui proses *matching* dan ekstraksi, yaitu seperti sistem – sistem yang ada di kumpulan *Carl-Way* menggunakan database contoh sebagai '*implicit knowledge*' selama *run-time*. Bahkan, penggunaan semacam itu tidak serta merta mendefinisikannya sebagai EBMT karena terjemahan dengan analogy '*translation by analogy*' juga bisa secara prinsip menjadi perluasan dari sistem transfer MT tradisional untuk memecahkan kasus – kasus dari ambiguitas lexical dimana tidak ada bukti langsung yang ditemukan dalam sebuah database terjemahan. Akibatnya, Turcato dan Popowitch memberikan kesimpulan bahwa



sebuah definisi yang tertutup dari EBMT adalah tidak penting. Hal yang paling penting adalah membuat sistem MT yang baik.

Bagaimanapun, ada 2 masalah utama dengan kesimpulan semacam itu. Pertama, itu tidaklah membantu peneliti dan juga peneliti MT yang lain jika dikatakan oleh praktisi EBMT sendiri bahwa tidak ada definisi yang jelas terhadap EBMT, padahal mereka perlu tahu bagaimana EBMT berbeda dari arsitektur MT yang lain. Kedua, pembatasan EBMT untuk penggunaan '*implicit knowledge*' pada *run-time* hanya akan terlihat sebagai suatu hal yang terlalu sempit, karena hal itu akan mengeluarkan banyak dari laporan penelitian dalam koleksi Carl-Way dan pada konferensi-konferensi saat ini. Di sisi lain, secara sederhana bisa dikatakan bahwa sebuah sistem adalah EBMT, hanya berdasarkan penulis artikel tersebut bukanlah sebuah jawaban.

### 2.3 Example-Based Machine Translation Arabic-English

Penelitian sebelumnya yang relevan dengan penerapan EBMT dalam penerjemahan bahasa Arab ke Bahasa Inggris adalah yang terdapat dalam paper . (KBar et al, 2007). Berikut ini dipaparkan rangkuman dari paper tersebut :

Tahapan –tahapan yang dilalui :

1. *Matching* : mencari *fragment* yang mirip.
2. *Transfer* : setelah didapatkan sekelompok *fragment* – *fragment* yang tepat, langkah selanjutnya adalah mengekstraksi terjemahan yang mungkin dari versi bahasa target (Bahasa Inggris) pada *corpus*.
3. *Recombination* : membangkitkan teks bahasa target yang lengkap

Implementasi yang dirancang adalah komponen utama dari system EBMT (*Example –Based Machine Translation*) yang menerjemahkan kalimat MSA (*Modern Standard Arabic*) ke bahasa Inggris. Sistem tersebut adalah non-struktural, yaitu menyimpan contoh terjemahan sebagai string tekstual, dengan tambahan informasi tentang morfologi dan *part-of-speech*.

Saat ini, sistem baru mampu memecah-mecah suatu kalimat inputan dan menterjemahkan masing-masing pecahan tersebut secara terpisah. Tahap

*recombinaton* menjadi gabungan kata –kata terjemahan tersebut secara utuh dan koheren akan dilakukan sebagai pekerjaan di masa mendatang.

Tujuan akhir adalah membangun sebuah asisten otomatis untuk Machine Translation System *Arabic-to-English* yang bekerja berdasarkan paradigma *rule-based* atau *statistic*, sehingga mampu menangani kasus-kasus rumit, khususnya untuk meningkatkan kelancaran terjemahan yang dihasilkan.

#### A. Menyimpan Contoh Terjemahan

Contoh – contoh terjemahan didapatkan dari kumpulan dokumen paralel *Arabic-English* yang diambil dari inventaris dokumen PBB yang tersedia dalam *Official –Document System (ODS)*.

Masing-masing parallel dokumen di-*align* pada level paragraf. Masing-masing parallel paragraf tersebut kemudian digunakan sebagai contoh terjemahan. Contoh-contoh ini kemudian secara morfologi dianalisis dengan *Buckwalter Morphological Analyser* versi 1, dan *part-of-speech* ditandai (*tagged*) dengan menggunakan SVM-POS. Dengan cara demikian, untuk masing-masing kata hanya dipertimbangkan berdasarkan hasil analisis *Buckwalter* dan SVM-POS yang relevan. Sebuah tabel *lookup* yang memetakan kata dalam Bahasa Arab ke kata dalam Bahasa Inggris-nya pada masing-masing parallel paragraf juga diciptakan secara otomatis. Untuk masing-masing kata Arab dalam contoh terjemahan, dilakukan *look-up* kata bahasa Inggris yang ekuivalen dan mencari sinonimnya menggunakan WordNet. Kemudian, dicarikan versi Bahasa Inggris untuk contoh terjemahan pada masing-masing *instance* di level *lemma* dan dimasukkan ke dalam tabel.

Versi Arab dalam *corpus* diindeks berdasarkan level kata, stem, dan lemma (stem dan lemma telah didefinisikan oleh *Buckwalter analyzer*), karena itu untuk masing-masing kata yang diberikan, dimungkinkan untuk mengambil (*retrieve*) seluruh contoh terjemahan yang berisi kata tersebut pada seluruh level.

#### B. Matching.

Ketika suatu input kalimat dimasukkan, system mulai mencari *corpus* pada contoh terjemahan yang versi Arab-nya sesuai dengan potongan-potongan (*fragments*) inputan. Potongan yang sesuai (*matched*) harus berisi paling tidak dua kata yang bersebelahan (*adjacent words*) pada inputan yang sama. Potongan yang sama bisa didapatkan pada lebih dari satu contoh terjemahan. Untuk keperluan tersebut, sebuah *match-score* diberikan pada masing-masing pasangan potongan terjemahan, yang menunjukkan kualitas potongan yang sesuai pada contoh terjemahan tertentu. Potongan-potongan (*fragments*) dicocokkan kata per kata, sehingga skor pada sebuah potongan adalah rata-rata dari tiap-tiap *match-score* kata yang membentuknya.

Kata-kata dicocokkan pada level *teks*, *stem*, *lemma*, dan *part-of-speech*, yang masing-masing level diberi skor yang berbeda. Teks (string yang persis sama) dan stem memberikan tingkat kecocokan yang paling tinggi, selanjutnya *lemma*, dan terakhir *part-of-speech*. Penjelasan tentang pemberian skor pada masing-masing level tersebut bisa dilihat pada tabel 2.14.

Kecocokan pada level teks dan stem menghasilkan skor tertinggi, karena system ini belum mampu mengatasi modifikasi penerjemahan. Pada teks (bahasa Arab) yang tidak berharakat, terdapat suatu hal yang rumit ketika kedua kata memiliki stem yang sama, tapi lemma yang berbeda. Contoh : kata *كتب* (*kataba*) berarti menulis, sedangkan *كتب* (*kutub*) berarti buku-buku. Kasus-kasus semacam ini belum bisa ditangani karena kita belum bekerja dengan sebuah *context sensitive Arabic lemmatizer* sehingga belum bisa menurunkan lemma yang tepat pada suatu kata berbahasa Arab. Selain itu, kombinasi dari *Buckwalter Morphological Analyzer* dengan SVM-POS tagger memungkinkan kita untuk mengurangi jumlah ambiguitas.

Dengan kecocokan lemma, berarti kata-kata sesuai dengan salah satu dari kemungkinan lemma yang ada. *Match-score* untuk kasus tersebut adalah perbandingan antara jumlah lemma yang cocok dengan total jumlah pasangan lemma (satu per kata). Penyelidikan lebih lanjut, sebagaimana membangun dan bekerja dengan *context sensitive arabic lemmatizer* masih dibutuhkan untuk mengatasi situasi-situasi semacam itu.

Potongan-potongan dengan skor di bawah *threshold* dibuang, karena melewati potongan-potongan yang berskor rendah pada tahap selanjutnya bisa menambah total *running time* secara drastis. Sebagai catatan, bahwa sebuah corpus yang lebih besar, dengan kecocokan yang bertambah sejumlah potongan-potongan potensial, akan membutuhkan peningkatan *threshold*.

Tabel 2.14 Skor Kecocokan pada Tiap Level

Tingkat	Penjelasan	Skor
Teks	Benar-benar cocok kata per kata	1
Stem	Sama stem-nya tapi bentuk permukaannya ( <i>surface form</i> ) tidak sama. Contoh :	0.9
	المستورية (konstitusional)	
	دمستوري (konstitusiku)	
	dua-duanya memiliki stem yang sama :	
	دمستوري	
Lemma	Contoh berikut memiliki lemma yang sama :	Dynamic score
	مارق	
	مراق	
Isi	Masih direncanakan, blm diimplementasikan	0.8
	Contoh : Untuk lokasi nama yang sama	
	memiliki skor yang lebih tinggi dibandingkan	
	2 kata benda yang tidak sama	
Part-of-Speech	Contoh : Dua-duanya adalah kata benda.	0.3
	Cocok dalam hal ma'rifat dan nakirahnya	
Common	Contoh : بـ dengan في dlm kelompok yg sama	0.1

Potongan-potongan disimpan dalam sebuah struktur yang terdiri dari :

- (1) *Source pattern* (pola sumber) : potongan teks Arabic yang diambil dari kalimat input.

(2) *Example pattern* (pola contoh) : potongan teks Arabic yang diambil dari contoh terjemahan yang cocok.

(3) *Example* (contoh) : terjemahan Bahasa Inggris dari pola contoh.

(4) *Match score* (skor kecocokan) dari potongan dan contoh terjemahannya.

Untuk efisiensi, potongan-potongan yang memiliki pola contoh sama dikumpulkan dan disimpan dalam level yang lebih tinggi, yaitu struktur *general-fragment*. (Perlu dicatat, bahwa mungkin juga sebuah *general-fragment* terdiri dari hanya 1 potongan saja).

### C. Transfer

Input dari tahapan transfer terdiri dari seluruh *general-fragment* yang ditemukan dalam tahapan *matching*, dan outputnya adalah penerjemahan dari seluruh *general-fragment* tersebut. Penerjemahan dari sebuah *general-fragment* tersebut diambil sebagai sebuah hasil penerjemahan terbaik di antara fragment-fragment anggotanya. Menerjemahkan sebuah fragment dilakukan dalam 2 langkah utama :

(1) Mengekstraksi penerjemahan dari pola contoh yang bersumber dari versi berbahasa Inggris dari contoh terjemahan.

(2) Memperbaiki penerjemahan yang telah diekstraksi sehingga bisa menjadi terjemahan dari potongan pola sumber (*source pattern*).

#### **Langkah pertama - Ekstraksi penerjemahan.**

Langkah pertama adalah mengekstraksi penerjemahan dari contoh *fragment* dari versi berbahasa Inggris. Dalam hal ini kita menggunakan tabel *look-up* untuk seluruh contoh penerjemahan dari corpus kita. Untuk seluruh kata berbahasa Arab dalam pola, kita melakukan *look up* terhadap bahasa Inggris yang ekuivalen di tabel dan menandainya dalam versi berbahasa Inggris pada contoh penerjemahan. Selanjutnya, kita mengekstraksi segmen berbahasa Inggris yang terpendek yang berisi maksimum jumlah kata – kata yang ekuivalen. Biasanya sebuah kata dalam beberapa contoh pola berbahasa Arab memiliki beberapa bahasa Inggris yang ekuivalen.

Sebagai contoh :

الخدمات الإستشارية والتعاون التقني في ميدان حقوق الإنسان



Terjemahan Bahasa Inggris : '*Advisory services and technical cooperation in the field of human rights*'

Table 2.15 adalah berisi look-up table yang bersesuaian. Coba kita perhatikan pola contoh berikut :

ميدان حقوق الإنسان

Terjemahannya adalah : '*the field of human rights*', jadi kita akan mengekstrak terjemahannya versi Bahasa Inggris yang ada di contoh penerjemahan. Dengan menggunakan table look-up yang telah diekstraksi, kita bisa menandai ekivalen Bahasa Inggris dari pola kata-kata yang ada di contoh penerjemahan : '*Advisory services and technical cooperation in the field of human rights*', dan kemudian kita mengekstrak segmen Bahasa Inggris yang paling pendek yang mengandung jumlah maksimum kata yang ekivalen, yaitu : "*field of human rights*". Itu tentu saja sebuah contoh yang sederhana. Untuk contoh yang lebih rumit seharusnya akan memiliki lebih dari satu ekivalen pada setiap kata berbahasa Arab.

Tabel 2.15 ALIGNMENT LOOK-UP TABLE

English	Arabic
Services	الخدمات
Advisory	الإستشارية
Cooperation	والتعاون
Technical	التقني
In	في
Field	ميدان
Rights	حقوق
Human	الإنسان

Kadangkala sulit untuk mencari padanan kata yang tepat dalam bahasa Inggris untuk sebuah kata tertentu berbahasa Arab. Biasanya hal itu terjadi ketika

kata berbahasa Arab itu adalah bagian dari beberapa frase, yang penerjemahannya tidak dilakukan persis kata per kata, contohnya :

غير رسمي

yang terjemahan Bahasa Inggrisnya adalah : *not formal*. Dalam beberapa keadaan, kita bisa saja menemukan terjemahan : '*informal*' sebagai terjemahan Bahasa Inggrisnya. Masalahnya adalah bahwa kata رسمي (*formal*), ataupun غير (*not*) sama – sama tidak berisi kata *informal*. Situasi semacam ini bisa ditangani dengan cara mendefinisikan aturan secara manual, yang dipicu ketika muncul kata غير , system seharusnya mengecek kata berikutnya, dan kemudian membangun list *antonym* (kata yang berlawanan) –bukan sinonimnya- dengan menggunakan WordNet. Pada contoh ini, kata *informal* muncul sebagai *antonym* dari kata *formal* di WordNet.

Terdapat struktur-struktur yang lebih rumit yang belum bisa ditangani, namun tampaknya penanganan semacam itu bisa dilakukan dengan menangkap dan menulis aturan tertentu yang sesuai untuk keadaan tersebut.

## 2) Langkah ke-dua--- Memperbaiki penerjemahan.

Perlu diingat kembali bahwa pada proses *matching* antara sebuah fragmen corpus dengan input fragmen bisa jadi tidak benar-benar tepat, yaitu kata-kata yang ada mungkin cocok pada beberapa level. Kata-kata yang memiliki kecocokan yang sama persis bisa diasumsikan memiliki terjemahan yang sama, tetapi kata-kata yang memiliki kecocokan dalam stem dan lemma akan membutuhkan modifikasi (kebanyakan masalah *inflection* dan preposisi) pada terjemahan yang diekstraksi. Masalah semacam ini dilewati sebagai garapan di masa datang. Kata-kata yang mengalami kecocokan pada level *part-of-speech* membutuhkan perubahan arti secara menyeluruh. Sebagai contoh, potongan input : مجلس الأمن (*the Security Council*) cocok dengan potongan : مسؤولية الأمن ('*the security responsibility*') pada beberapa contoh terjemahan. Kata مجلس (*council*)

dan مسؤولية (*responsibility*) adalah cocok pada level *part-of-speech* (kedua-duanya adalah kata benda). Diasumsikan bahwa terjemahan yang diekstraksi dari contoh terjemahan adalah "*the security responsibility*", yang merupakan terjemahan yang tepat dari مسؤولية الأمن dan bukan merupakan terjemahan dari pola input secara keseluruhan. Tapi, dengan cara menghilangkan kata *responsibility* dari contoh terjemahan dengan terjemahan dari مجلس (*council*) yang diambil dari *lexicon* (kamus). *Lexicon* tersebut diambilkan dari daftar kata yang diekstraksi dari *Buckwalter Morphological Analyser* dan dikembangkan dengan synonym pada WordNet sebagaimana dijelaskan di atas.

Kadangkala terjemahan yang diekstraksi berisi beberapa kata-kata tambahan di tengah yang tidak diperlukan. Kata-kata tersebut biasanya muncul karena perbedaan struktur dari frase kata benda pada kedua bahasa. Sebagai misal, perhatikan contoh : موضوع الأمن الأقليمي dan terjemahannya : "*the subject of regional security*". Dengan mengekstraksi terjemahan dari pola موضوع الأمن , kita dapatkan : "*the subject of regional security*" (karena itu adalah segmen terpendek yang berisi jumlah maksimum *word alignment*/ kata dua bahasa yang ekuivalen). Jelasnya, kata "*regional*" tidak dibutuhkan karena itu adalah terjemahan terhadap kata : الأقليمي. Sehingga, dengan cara menghilangkan kata tersebut dari terjemahan, kita dapatkan terjemahan yang benar dalam pola tersebut. Kata "*regional*" muncul pada terjemahan yang diekstraksi karena kata sifat dalam bahasa Arab terletak setelah kata benda yang sesuai. Hal itu berlawanan dengan sintaks Bahasa Inggris. Dalam hal ini, frase kata benda الأمن الأقليمي ("*the regional security*") diterjemahkan sehingga terjemahan dari الأقليمي muncul sebelum terjemahan dari الأمن. Saat ini, untuk mengidentifikasi situasi semacam itu bisa dilakukan dengan cara mencari terjemahan dari kata "*regional*" pada sejumlah kata berbahasa Arab yang muncul secara langsung setelah pola yang ada pada contoh terjemahan. Bagaimanapun, cara ini tidak cukup untuk situasi yang lebih kompleks dan juga sangat memakan banyak waktu. Rencana kami

selanjutnya adalah mengaplikasikan sebuah Arabic *chunker* untuk mengekstraksi batasan-batasan dari frase kata benda, sehingga dengan cara tersebut mengurangi area pencarian.

Menghilangkan kata-kata yang tidak perlu dari terjemahan yang diekstraksi harus mempertahankan sintaks Bahasa Inggris yang benar dari sisa terjemahan, - yang dalam beberapa kasus – terlihat sebagai sebuah pekerjaan yang sulit. Untuk tujuan itu, kami telah menyusun beberapa aturan (*rules*) untuk menangani situasi yang berbeda. Aturan-aturan ini berdasar pada sintaks Bahasa Inggris pada terjemahan yang diekstraksi dan mengidentifikasi kasus-kasus yang membutuhkan penanganan khusus. Pertama, kami memotong terjemahan untuk mencari frase kata benda dasar, dengan menggunakan BaseNP *chunker*. Untuk melakukan hal itu, sebelumnya kami mengaplikasikan *Brill's part-of-speech tagger* pada terjemahan. Selanjutnya, dengan memperhatikan teks Bahasa Inggris yang terpotong (*chunked*) , kami bisa memastikan pengaruh penghilangan kata yang tidak perlu. Pada contoh sebelumnya, menghilangkan kata "*regional*" dari teks : "*the subject of regional security*" bisa dilakukan tanpa modifikasi lebih lanjut, karena dengan proses *tagging* dan *chunking* terhadap segment, kita bisa mendapatkan :

[the/DT subject/NN] of/IN [regional/JJ security/NN] (frase yang berada di dalam kurung adalah frase kata benda) dan "*regional*" adalah kata sifat dengan sebuah frase kata benda yang tetap memiliki kepala (*head*) yang sama. Preposisi dan kata fungsi yang lain yang berhubungan dengan frase masih tetap dibutuhkan, sehingga kami tetap mempertahankannya.

Sebagaimana telah disebutkan sebelumnya, sebuah *general-fragment* bisa jadi berisi beberapa fragment yang memiliki contoh pola Bahasa Arab yang sama. Di antara terjemahan –terjemahan yang diekstraksi dari anggota fragment, yaitu seluruh terjemahan dari pola Bahasa Arab yang sama, kami memilih terjemahan yang mencakup jumlah maksimum kata Bahasa Arab untuk merepresentasikan *general-fragment*. Skor terjemahan dari terjemahan yang terpilih adalah perbandingan antara jumlah kata-kata yang tercakup dengan total jumlah kata

dalam pola Bahasa Arab. Skor total dari sebuah *general-fragment* adalah perkalian dari skor ketepatan (*match-score*) dengan skor terjemahan.

#### D. *Recombination*.

Pada tahapan *recombination*, peneliti memasangkan secara bersamaan terjemahan yang diekstraksi untuk membentuk sebuah terjemahan lengkap dari sebuah kalimat inputan. Secara umum tahapan ini terdiri dari dua sub pekerjaan. Pertama, mencari N jumlah rekombinasi terbaik dari terjemahan yang diekstraksi yang mencakup seluruh kalimat input, dan yang kedua adalah menghaluskan terjemahan rekombinasi untuk menghasilkan sebuah kalimat berbahasa Inggris yang tepat secara gramatikal. Saat ini, penulis paper hanya menangani hanya sub pekerjaan yang pertama, sedangkan yang kedua disisihkan sebagai garapan selanjutnya di masa mendatang. Dengan mengalikan skor total anggota *general-fragment*, dapat dihitung sebuah skor final terjemahan untuk masing-masing rekombinasi yang dihasilkan. N rekombinasi terbaik kemudian dilaporkan.



**BAB 3**  
**COMPUTER AIDED TRANSLATION**  
**ARAB-INDONESIA**

### **BAB 3**

## **COMPUTER AIDED TRANSLATION**

### **ARAB-INDONESIA**

#### **3.1 Pengertian Computer Aided Translation (CAT)**

*Computer Aided Translation (CAT)* adalah suatu system computer yang digunakan sebagai piranti bantu dalam proses penerjemahan. Berbeda dengan *machine translation (MT)*, CAT memberikan peluang akhir bagi penerjemah (manusia) untuk menentukan output akhirnya. CAT hanyalah melakukan asistensi dan memberikan dukungan untuk dihasilkan penerjemahan terbaik. CAT juga sering disebut sebagai *Human Aided Machine Translation (HAMT)*.

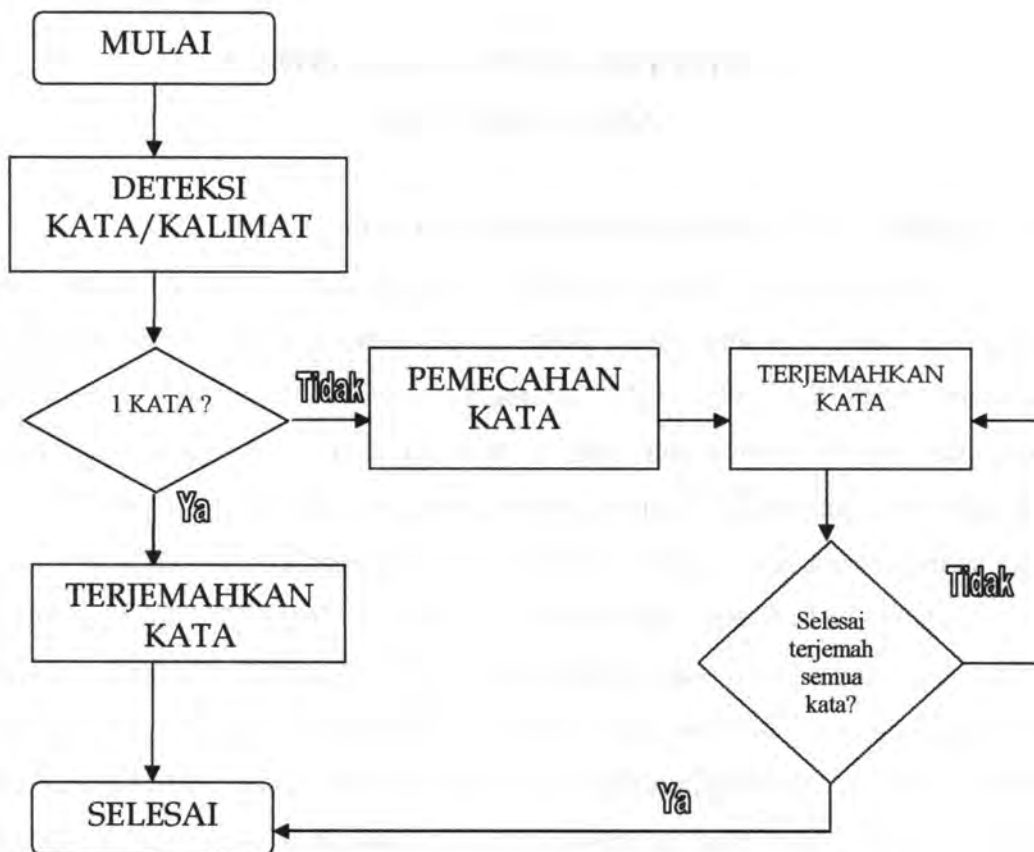
Dalam MT dikenal adanya *Fully Automatic High Quality Translation (FAHQT)*, yaitu system yang diharapkan secara langsung menghasilkan output terjemahan dengan kualitas tinggi. Dalam penerjemahan bahasa Arab, masih banyak ditemui kekurangan dalam menerapkan FAHQT tersebut. Karena itu, system terbaik yang bisa digunakan saat ini khususnya yang terkait dengan penerjemahan bahasa Arab adalah CAT atau HAMT( Al-Wasiti, 2005).

#### **3.2 Perancangan CAT Arab-Indonesia**

##### **3.2.1 Alur Utama Proses Penerjemahan**

Sistem yang dirancang tidak hanya mampu menerjemahkan inputan satu kata menjadi satu output kata terjemahan, namun juga diharapkan bisa mengakomodasi inputan suatu frase atau kalimat. Karena itu, system bisa mendeteksi apakah inputannya tersebut adalah 1 kata atau lebih dari 1 kata. Jika ternyata lebih dari satu kata, masing-masing kata akan dipecah untuk diterjemahkan.

Pada Gambar 3.1 terlihat bahwa pada saat terjadi pemecahan kata, masing-masing kata tersebut akan diterjemahkan. Proses penerjemahan berlangsung secara berulang sampai seluruh kata dicek. Jika semua kata telah mengalami proses penerjemahan, barulah proses dianggap telah berakhir. Suatu kata dianggap telah mengalami proses penerjemahan meskipun kata tersebut tidak didapati dalam cakupan database sistem. Jika tidak didapati adanya kata yang sesuai,



Gambar 3.1 Diagram Alur Utama Proses Penerjemahan

termasuk dengan proses *tashrif* yang akan dijelaskan pada Bab 4, sistem akan menghasilkan keluaran string '['].

### 3.2.2 Dukungan yang Diharapkan dari Sistem

Sebagai piranti bantu dalam proses penerjemahan, sistem diharapkan bisa melakukan hal-hal sebagai berikut:

1. Menerjemahkan setiap kata yang diinputkan.
2. Mengakomodasi inputan tanpa *harakat*.

Mayoritas teks bahasa Arab ditampilkan tanpa *harakat* (tanda baca vokal), karena itu sistem harus bisa mengakomodasi inputan tanpa *harakat*.

3. Menampilkan pensejajaran terjemahan masing-masing kata pada *user*.

Misalkan, jika diinputkan kalimat : ذهب خالد إلى السوق, selain ditampilkan output terjemahan kalimat secara utuh: 'pergi Kholid ke



pasar', system bisa menampilkan *lookup alignment* terjemahan masing-masing kata seperti dalam Tabel 3.1.

Tabel 3.1 Contoh *LookUp Alignment* Masing-masing Kata Inputan

Arti	Kata
pergi	ذهب
Kholid	خالد
Ke	إلى
pasar	السوق

4. Menampilkan kata-kata yang memiliki kemungkinan banyak makna. Pada Tabel 3.1 kata خالد bisa berarti nama orang: Kholid, atau juga bisa bermakna 'kekal'. Dua kemungkinan kata ini bisa ditampilkan pada *user* dengan harakatnya jika *user* mengharapkan asistensi lebih lanjut.
5. Memilih kata yang lebih tepat terhadap kata yang banyak makna tersebut. Setelah ditampilkan kemungkinan makna lain dan ternyata *user* cenderung pada makna lain, maka *user* diberi akses secara langsung dan interaktif untuk merubah pilihannya.

### 3.2.3 Menangani Input Tanpa Harakat

Untuk menangani inputan tanpa *harakat*, setiap isian dalam database harus terdiri dari field yang berharakat dan field yang tidak berharakat. Sebagai contoh, untuk kata benda dengan terjemahannya, perlu tambahan kolom untuk menampung index kata tidak berharakat seperti dalam Tabel 3.2.

Kolom 'Non Harakat' pada Tabel 3.2 merupakan representasi dari kata dalam kolom 'Arab' dengan menghilangkan harakatnya. Harakat dalam bahasa Arab adalah tanda baca vokal yang berada di atas atau di bawah suatu huruf. Harakat yang dikenal ada 8: *fathah* (ـَ) sebagai tanda baca vokal 'a', *dhommah* (ـُ) sebagai tanda baca vokal 'u', *kasroh* (ـِ) sebagai tanda baca vokal 'i',

*fathatain* (ـَ) sebagai tanda baca vokal 'an', *dhommatain* (ـٌ) sebagai tanda baca vokal 'un', *kasrotain* (ـِ) sebagai tanda baca vokal 'in', *sukun* (ـْ) sebagai tanda baca 'mati', dan *tasydid* atau *syaddah* (ـّ) sebagai tanda baca yang menyebabkan pembacaan suatu huruf dibaca ganda (pembacaan huruf awal di'hidupkan' sedangkan huruf kedua dibaca 'mati').

Tabel 3.2 Contoh Penyimpanan Kata Benda, Terjemahan dan Index Non Harakat

Arti	Arab	Non Harakat
Pergi	ذَهَبَ	ذهب
Pasar	سُوقٌ	سوق
masjid	مَسْجِدٌ	مسجد

Sebelum dimasukkan dalam database, kata berharakat harus dibersihkan dari harakatnya dengan menggunakan kode program. Contoh kode program untuk membersihkan harakat dari suatu kata dalam Visual Basic 6.0 dapat dilihat pada Gambar 3.2.

```

t1 = kata_arab
pt1 = Len(t1)

tString = ""
For k = 1 To pt1
  If Trim(Mid(t1, k, 1)) <> "" And Trim(Mid(t1, k, 1)) <> ""
    And Trim(Mid(t1, k, 1)) <> "" And Trim(Mid(t1, k, 1)) <> ""
    And Trim(Mid(t1, k, 1)) <> "" And Trim(Mid(t1, k, 1)) <> ""
    And Trim(Mid(t1, k, 1)) <> "" Then
    tString = tString & Mid(t1, k, 1)
  End If
Next
Non_harakat = tString

```

Gambar 3.2 Kode VB6 untuk menghilangkan harakat pada kata

Kata berbahasa Arab yang mengandung harakat disimpan dalam variabel 'kata\_arab'. Selanjutnya, dilakukan penghitungan panjang string kata berharakat tersebut. Disiapkan sebuah variabel 'tString' untuk menampung hasil akhir kata yang telah dibersihkan dari harakatnya. Variabel tString itu diinisialisasi dengan string kosong terlebih dahulu. Kemudian, dilakukan proses perulangan untuk mendeteksi character 'kata\_arab', selama character tersebut bukan harakat, maka dia akan disambungkan dengan character sebelumnya yang telah disimpan dalam variabel tString. Hasil akhirnya, variabel tString akan berisi string kata yang telah bersih dari harakat.

### 3.2.4 Menyiapkan Struktur Database

Database digunakan dalam CAT yang dirancang untuk acuan kamus kata terjemahan. Perlu dilakukan pengaturan struktur tabel dalam database sedemikian sehingga didapatkan mekanisme penyimpanan yang minim *redundansi* data, efektif (menampung semua atribut data yang diperlukan), serta efisien (hemat *space* harddisk).

Kata yang akan diinputkan bisa berupa kata kerja atau kata benda. Kedua jenis kata tersebut memerlukan penanganan tersendiri karena karakteristiknya berbeda. Kata benda bisa mengalami partikel sisipan 'U', sedangkan kata kerja tidak. Kata benda dalam bahasa Arab juga bisa mengalami partikel akhiran 'l' jika dia mengalami keadaan *nashob*.

Kata kerja memerlukan mekanisme penyimpanan yang khusus karena merupakan obyek/ sasaran dalam proses *tashrif* (seperti akan dijelaskan dalam Bab 4). Karena itu, kata kerja dan kata benda tidak bisa digabungkan dalam satu tabel.

Mekanisme penyimpanan yang tidak kalah penting lagi adalah penyimpanan contoh terjemahan. Tabel contoh terjemahan menampung kalimat/ frase dalam bahasa Arab dan terjemahannya dalam bahasa Indonesia. Tabel contoh terjemahan tersebut membutuhkan tabel pendamping yang memetakan terjemahan seluruh kata dalam bahasa Arab ke bahasa Indonesia yang bersesuaian. Hubungan antara tabel contoh terjemahan dengan tabel pendamping

tersebut adalah 'Master-Detail'. Tabel pendamping tersebut bisa dinamakan sebagai *word alignment* sesuai dengan fungsinya.

Untuk menangani kata yang bermakna banyak (*polysemi*) atau kata yang memiliki sinonim banyak, perlu dibuatkan tabel tersendiri. Karena itu dibutuhkan tabel sinonim yang menampung keadaan kata yang demikian.

Masing-masing tabel tersebut tidak saling berhubungan kecuali tabel Contoh Terjemahan dengan Word Alignment. Hal ini terjadi karena proses pencariannya secara fisik tidak terkait antar satu tabel dengan tabel yang lain, namun secara logik ada keterkaitan. Jika di dalam suatu tabel tidak ditemukan kata yang dimaksud, maka akan dicari pada tabel selanjutnya, dan seterusnya sesuai dengan urutan prioritas.

### 3.2.5 Struktur Tabel dalam Database

Setelah dilakukan proses analisis kebutuhan penyimpanan, dapat dirancang struktur tabel masing-masing dalam database.

#### Contoh Terjemahan

Contoh terjemahan suatu kalimat atau frase tersimpan dalam tabel 'Contoh' dan didetailkan pada tabel 'Word\_alignment' untuk terjemahan masing-masing kata.

Tabel 3.3 Struktur Tabel Contoh

Nama Field	Tipe	Lebar
id_kal	Number	Integer
Arab	Text	200
Indonesia	Text	200

Field *Id\_kal* adalah nomor identifikasi tiap kalimat dan merupakan kunci primer (*primery key*). Field *Arab* berisi kalimat dalam bahasa Arab, sedangkan field *Indonesia* berisi data terjemahan kalimat dalam bahasa Indonesia.

Tabel 3.4 Struktur Tabel Word\_alignment

Field	Tipe	Lebar
id_kal	Number	Integer
id_kata	Number	Integer
Arab	Text	40
Indonesia	Text	40

Field id\_kal adalah nomor identifikasi tiap kalimat yang merupakan kunci asing (*foreign key*). Id\_kata adalah nomor urut kata tersebut dalam kalimat. Field arab berisi kata dalam bahasa Arab yang diterjemahkan dalam bahasa Indonesia pada field Indonesia. Untuk pengembangan selanjutnya, bisa ditambahkan field non\_harakat berisi kata pada field arab yang telah bersih dari harakat, sedangkan field 'Arab' diisi untuk kata yang memiliki harakat.

Keterhubungan dua tabel tersebut dapat dilihat dalam Gambar 3.3:



Gambar 3.3 Keterhubungan tabel Contoh dan Word\_alignment

Field yang menghubungkan antara tabel Contoh dengan tabel Word\_alignment adalah id\_kal. Meskipun pada tabel contoh dan word\_alignment sama-sama memiliki field arab dan indonesia, namun lebar data yang diisi berbeda, karena tabel contoh berorientasi pada kalimat/ frase sedangkan tabel word\_alignment berorientasi pada kata.

### Daftar Kata Benda

Daftar kata benda beserta terjemahannya disimpan dalam tabel sendiri yang

disebut tabel 'jama'. Tabel ini menampung kata benda dalam bentuk tunggal maupun dalam bentuk jama'. Tabel 3.5 menunjukkan struktur tabel tersebut.

Tabel 3.5 Struktur Tabel 'Jama' sebagai Penyimpan Daftar Kata Benda

Field	Tipe	Lebar
Arti	Text	60
Jama	Text	25
Tunggal	Text	25
Gjama	Text	25
Gtunggal	Text	25

Field 'arti' menampung data arti bahasa Indonesia dari kata benda. Field jama menampung kata benda jama untuk kata benda dalam bahasa Arab beserta harakatnya, sedangkan kata benda jama tanpa harakat tersimpan dalam field 'gjama'. Field 'tunggal' menampung data kata benda tunggal dalam bahasa Arab dengan harakatnya, sedangkan field 'gtunggal' menampung kata benda tunggal yang telah dibersihkan dari harakatnya. Kata benda yang disimpan dalam tabel ini berbentuk *makrifah*, tidak mengandung partikel 'ا'.

### List Kata Kerja

Daftar kata kerja disimpan dalam tabel tersendiri yang dinamakan 'induk1'. Tabel 3.6 menunjukkan struktur tabel dari tabel tersebut.

Tabel 3.6 Struktur Tabel 'Induk1' sebagai List Kata Kerja

Field	Tipe	Lebar
Non_harakat	Text	10
Arab	Text	15
Indonesia	Text	15
Kd_jenis	Number	Integer
Kd_wazan	Number	Integer
Fa	Text	1
Ain	Text	1
Lam	Text	1

Field 'non\_harakat' berisi kata kerja dalam bahasa Arab tanpa harakat yang sumbernya adalah kata kerja yang tersimpan dalam field 'Arab'. Field 'Indonesia' berisi data terjemahan kata kerja tersebut dalam bahasa Indonesia. 'Kd\_jenis'

menunjukkan jenis kata tersebut sebagaimana penomorannya disebutkan dalam Tabel 2.7. 'Kd\_wazan' menunjukkan wazan dari kata kerja tersebut sebagaimana penomorannya disebutkan dalam Tabel 2.6. Field 'fa' menyimpan huruf pertama dalam field 'Non\_harakat' atau *fa' fi'il* dari kata kerjanya, sedangkan field 'ain' menyimpan huruf ke-2 field 'non\_harakat'. Field 'lam' menyimpan huruf ke-3 dari field 'non\_harakat' atau *lam fi'il* dari kata kerja. Kata kerja yang diinputkan dalam tabel ini adalah kata kerja bentuk lampau (*fi'il madhi*) untuk kata ganti orang ke-3 jenis laki-laki.

### Polysemi dan Sinonim Kata

Untuk menampung kata-kata benda yang memiliki arti banyak, dan memiliki sinonim dengan kata yang lain dibuatkanlah tabel tersendiri. Tabel tersebut dinamakan 'sinonim'. Dalam istilah bahasa Arab, kata tersebut mengalami *muroodif*.

Contoh kata yang mengalami keadan demikian adalah kata *عِلَّة*. Kata ini bisa berarti 'alasan' yang sinonim dengan kata *حُجَّة*. Kata *عِلَّة* juga bisa bermakna 'udzur' yang sinonim dengan kata *عُذْر*.

Keadaan sinonim ini perlu disimpan dalam tabel sebagai referensi CAT untuk menampilkan kata lain yang memiliki sinonim tertentu jika *user* membutuhkan penjelasan tambahan. Tabel 3.7 menunjukkan struktur tabel penyimpanan database untuk sinonim.

Tabel 3.7 Struktur Tabel Sinonim

Field	Tipe	Lebar
Kata	Text	10
Sinonim	Text	10
Arti	Text	15
har_kata	Text	8
Har_sin	Text	8

Field 'Kata' menyimpan data kata benda dalam bahasa Arab yang berharakat, sedangkan sinonimnya disimpan dalam field 'sinonim' juga dalam keadaan berharakat. Field 'har\_kata' menyimpan data pada field 'kata' yang tidak

mengandung harakat, sedangkan field 'har\_sin' menyimpan data pada field 'Sinonim' yang sudah bersih dari harakat. Data yang dimasukkan dalam tabel ini adalah untuk kata benda saja, karena sinonim kata pada

### 3.2.6 Proses Memecah Kalimat Menjadi Kata-kata Penyusunnya

Untuk memecah suatu kalimat menjadi kata-kata penyusunnya, dibutuhkan sebuah fungsi (*function*). Fungsi ini akan selalu dipanggil jika dibutuhkan proses pemecahan kalimat menjadi kata-kata penyusunnya. Dalam pengembangan CAT ini fungsi tersebut dimanfaatkan dalam 2 keadaan:

1. Entry 'contoh terjemahan'.  
Pada saat dientrykan contoh terjemahan suatu kalimat, selain terjadi proses penyimpanan ke tabel 'Contoh', juga dilakukan penyimpanan di tabel 'Word Alignment' untuk masing-masing kata penyusunnya.
2. Pendeteksian inputan sebagai kata atau kalimat, serta penerjemahan masing-masing kata penyusun kalimat.

Gambar 3.4 menunjukkan kode program dalam bahasa Visual Basic 6.0 untuk fungsi pemecahan suatu kalimat tersebut.

Nama fungsinya adalah SplitString dengan cakupan 'Public' sehingga bisa dipanggil pada seluruh *procedure*. Parameter fungsi ini terdiri dari string kalimat yang akan dipecah (TheString\$), *delimiter/character* pemisah (Delim\$), dan array tempat menampung hasil pemecahan(DynArray\$). Nilai yang dihasilkan dari fungsi ini adalah sebuah integer yang merepresentasikan jumlah kata penyusun kalimat.

Misalkan string kalimat yang akan dipecah tersimpan dalam variabel 'Text\$', delimiter yang akan digunakan adalah spasi kosong (" "), dan array tujuan adalah 'Words\$', dan keluaran dari fungsi ditampung dalam variabel 'n'. Untuk memanggil fungsi tersebut bisa dijabarkan dalam baris kode berikut:

```
N = SplitString(Text$, " ", Words$())
```



```

Public Function SplitString%(TheString$, Delim$, DynArray$())
  Dim p%, t%
  Dim sTempString$, tmp$
  t% = 0
  ' Remove trailing blanks
  sTempString$ = Trim(TheString$)
  ' If the last character is the delimiter, remove it
  If Right(sTempString$, 1) = Delim$ Then sTempString$ =
Left(sTempString$, Len(sTempString$) - 1)
  ' Add a delimiter to end
  sTempString$ = sTempString$ & Delim$
  Do
    p% = InStr(sTempString$, Delim$)
    If p% = 0 Then Exit Do
    tmp$ = Left(sTempString$, p% - 1)
    sTempString$ = Right(sTempString$, Len(sTempString$) - p%)
    DynArray$(t%) = Trim(tmp$)
    t% = t% + 1
  Loop
  ' Returns the last available index
  SplitString% = UBound(DynArray$)
End Function

```

Gambar 3.4 Fungsi Pemecah Kalimat Menjadi Kata Penyusunnya

Jika inputannya adalah 1 kata, maka nilai  $n=0$ . `Words$` adalah *array* dinamis yang menampung kata-kata penyusun kalimat. Index awal array tersebut dihitung dari 0. Proses ekstraksi masing-masing kata bisa dilakukan dengan melakukan perulangan pengambilan data nilai pada *array* `Words$` dari 0 sampai  $n$ .

### 3.2.7 Ekstraksi Huruf Penyusun pada Kata Kerja

Pada Tabel 3.6 terdapat field-field yang mewakili huruf penyusun suatu kata kerja (*fi'il madhi*). Field-field tersebut adalah *fa*, *ain*, dan *lam*, yang merepresentasikan huruf *fa*, *'ain*, dan *lam* *fi'il* dari suatu *fi'il madhi*. Posisi dari huruf-huruf tersebut berbeda-beda sesuai dengan *template wazan* masing-masing.

Posisi masing-masing huruf pada tiap template wazan dijabarkan dalam Tabel 3.8. Penomoran wazan di Tabel 3.8 mengikuti penomoran pada Tabel 2.6.

Tabel 3.8 Posisi Huruf Pembentuk *Fi'il Madhi* bina' *shohih* pada *Template Wazan*

No wazan	Fa'	Ain	Lam
1, 2, 3, 4, 5, 6, 14	1	2	3
16	2	3	4
15	1	3	4
18	2	3	4
17	2	5	6
19	2	4	5
20	3	4	5
22	4	5	6

Angka di bawah kolom Fa', Ain, dan Lam menunjukkan posisi huruf dalam *fi'il madhi*. Posisi tersebut sesuai dengan posisi dalam *fi'il madhi* tanpa harakat. Untuk wazan no.1 yaitu *فَعَلَ-يَفْعُلُ* *fa' fi'ilnya* ada pada posisi 1, *ain fi'il* pada posisi ke-2, dan *lam fi'il* pada posisi ke-3. Contoh untuk wazan ini adalah kata kerja *نَصَرَ* (telah menolong). *Fa' fi'ilnya* adalah huruf ن (nun) karena terletak pada posisi ke-1, *'ain fi'ilnya* adalah huruf ص (shad) karena terletak pada posisi ke-2, dan *lam fi'ilnya* adalah huruf ر (ra') karena terletak pada posisi ke-3. Aturan posisi masing-masing huruf pembentuk seperti ini juga berlaku pada wazan no 2,3,4,5,6,dan 14. Pada Tabel 3.8 terlihat bahwa *fa' fi'il* tidak selalu pada posisi ke-1, *ain fi'il* tidak selalu di posisi ke-2, dan *lam fi'il* tidak selalu berada pada posisi ke-3.

Sebagai contoh, bisa dilihat wazan no 22 yaitu *اسْتَفْعَلَ-يَسْتَفْعِلُ* posisi *fa' fi'il* adalah nomor 4, *ain fi'il* nomor 5, dan *lam fi'il* nomor 6. Contoh kata *fi'il madhi* yang memenuhi wazan ini adalah *اسْتَغْفَرَ* (memohon ampunan). *Fa' fi'ilnya* adalah غ (ghain) karena berada pada posisi ke-4 setelah kata tersebut dibersihkan dari harakatnya, *ain fi'ilnya* adalah ف (fa') karena berada pada posisi ke-5, dan *lam fi'ilnya* adalah ر (ra') karena terletak pada posisi ke-6.

Contoh pada Tabel 3.8 adalah untuk bangunan kata *shahih*, yaitu yang tidak mengandung huruf *illat* (ﻯ, ﺀ, dan ﺀ). Aturan-aturan lain bisa ditambahkan untuk bangunan kata yang lain dengan mengacu pada wazan masing-masing. Acuanya adalah pada wazan, bukan pada contoh kata karena langsung merepresentasikan tampilan huruf fa', ain, dan lam fi'il pada posisi masing-masing. Aturan-aturan (*rules*) tersebut bisa disimpan dalam database pada sebuah tabel sebagai acuan dalam mengekstrak template pada *tashrif system*. Penjelasan tentang template di *tashrif system* akan dijelaskan di Bab 4.

Jika disimpan dalam sebuah tabel di database, perlu dilakukan modifikasi terhadap layout kolom dan baris di Tabel 3.8. Tabel di database itu juga harus bisa menampung bangunan kata lain selain *shahih*. Acuan penomoran jenis bangunan kata bisa mengacu pada Tabel 2.12. Penyesuaian modifikasi kolom dan baris agar bisa diterapkan dalam mekanisme penyimpanan database adalah seperti Tabel 3.9.

Tabel 3.9 Penyesuaian Struktur Tabel di Database  
untuk Rule Huruf Pembentuk *Fi'il Madhi*

No wazan	Fa'	Ain	Lam	No Bangunan Kata
1	1	2	3	1
2	1	2	3	1
3	1	2	3	1
4	1	2	3	1
5	1	2	3	1
6	1	2	3	1
14	1	2	3	1
16	2	3	4	1
15	1	3	4	1
18	2	3	4	1
17	2	5	6	1
19	2	4	5	1
20	3	4	5	1
22	4	5	6	1

Tabel 3.9 menunjukkan penyesuaian struktur kolom dan baris dari Tabel 3.8 agar bisa ditransformasikan pada penyimpanan database. No wazan

1,2,3,4,5,6, dan 14 memiliki aturan yang persis sama, karena itu berada dalam 1 baris pertama di Tabel 3.8. Namun, jika akan disimpan dalam database, penyimpanan tiap item data haruslah atomik, sehingga dipisah menjadi 7 baris. Hasil modifikasi itu nampak pada baris ke 1 sampai 7 di Tabel 3.9.

### 3.2.8 Proses Pencocokan Input Kata dengan Database

Pencocokan kata inputan dengan database meliputi beberapa tahap pencarian:

1. Tabel contoh terjemahan.

Tabel contoh terjemahan berisi kalimat non harakat dalam bahasa Arab dengan terjemahannya dalam bahasa Indonesia. Jika inputan adalah berupa frase/ kalimat, inputan itu akan dicocokkan dengan baris yang berisi contoh frase/ kalimat yang sudah ada. Tabel 3.10 menunjukkan contoh isi data pada tabel 'Contoh'.

Tabel 3.10 Contoh Isi Data pada Tabel 'Contoh'

id_kal	arab	indonesia
1	استغفر الله العظيم	aku meminta ampun kepada Allah Yang Maha Agung
2	إنا لله وإنا إليه راجعون	Sesungguhnya kami adalah milik Allah dan kepadaNya kami kembali
6	الله ما في السموات وما في الأرض	milik Allahlah saja segala yang di langit dan segala yang di bumi
7	ذهب حامد إلى المدرسة	Hamid pergi ke sekolah
8	الله الغفار	Allah Yang Maha Pengampun
9	الطرس الورقة المكتوب فيها	Kertas adalah lembaran yang tertulis padanya
10	تسقط الورقة من الشجرة	daun jatuh dari pohon
11	ينام عباس فوق السرير	Abbas tidur di atas tempat tidur
12	كتب المدرس الدرس على السبورة	guru menulis pelajaran pada papan tulis
15	إذا بلغ الماء قطنين لم ينس	jika air mencapai dua kullah tidak najis

Isi data pada Tabel 3.10 bisa berupa kalimat utuh, bisa pula sebuah frase. Id\_kal 8 merupakan contoh frase, sedangkan id\_kal yang lain adalah kalimat utuh. Penerjemahan yang dilakukan adalah berorientasi pada kalimat, bukan kata per kata.

Jika memang ada baris yang cocok, persis sama tiap character, maka terjemahan dalam bahasa Indonesianya ditampilkan sebagai output. Jika tidak ditemukan, maka dilanjutkan pada tahap selanjutnya, yaitu nomor 2: tabel word\_alignment.

## 2. Tabel Word\_alignment.

Tiap baris pada tabel Word\_alignment berisi terjemahan kata dari suatu kalimat tertentu. Tabel word\_alignment merupakan detail dari tabel 'Contoh'. Sebagai contoh, Tabel 3.11 menunjukkan isi tabel Word\_alignment untuk Id\_kal 11.

Kata yang ada dalam tabel Word\_alignment diterjemahkan kata per kata. Orientasinya adalah kata, bukan seperti di tabel Contoh yang berorientasi pada kalimat. Kata yang disimpan adalah kata penyusun dari frase atau kalimat, sehingga bisa berupa kata benda atau kata kerja. Kata benda pada Tabel 3.11 di antaranya ada pada ID\_KATA 2,3, dan 4, sedangkan kata kerjanya ada pada ID\_KATA 1. Jika kata yang diinputkan tidak terdapat pada tabel Word\_alignment, maka pencarian dilakukan pada tahap selanjutnya, tahap 3: Tabel Jama.

Tabel 3.11 Isi Tabel Word\_alignment untuk Id\_kal 11

ID_KAL	ID_KATA	ARAB	INDONESIA
11	1	ينام	Sedang tidur
11	2	عباس	Abbas
11	3	فوق	di atas
11	4	السريير	tempat tidur

## 3. Tabel Jama

Tabel Jama berisi kata benda tunggal dan kata benda jama' (lebih dari 2). Contoh isi data pada tabel Jama adalah seperti ditunjukkan Tabel 3.12

Acuan pencarian kata tidak berharakat untuk bentuk tunggal adalah field Gtunggal, sedangkan untuk kata jama' adalah Gjama. Semua kata yang diisikan pada tabel Jama berbentuk. Tidak semua kata tunggal memiliki jama', seperti pada kata إِعَادَةٌ dan إِطْلَاقٌ.

Proses pencarian pada tabel Jama meliputi beberapa tahap:



- a) Pencarian di kolom kata tunggal, jika tidak ditemukan yang persis sama character demi character, maka dilakukan penyesuaian. Penyesuaian tersebut dilakukan dengan menghilangkan partikel awalan dan akhiran.

Tabel 3.12 Contoh Isi Data Tabel Jama

Arti	Jama	Tunggal	Gjama	Gtunggal
Dokter	أَطِبَّاءُ	طَبِيبٌ	أطباء	طبيب
Pentelaahan		إِطْلَاعٌ		إِطْلَاعٌ
anak kecil	أَطْفَالٌ	طِفْلٌ	أطفال	طفل
pengembalian : pengulangan		إِعَادَةٌ		إعادة
Bapak	أَبَاءٌ	أَبٌ	أباء	أب
Sumur	أَبَارٌ	بَيْتْرٌ	أبار	بئر
peninggalan, bekas	أَثَارٌ	أَثْرٌ	أثار	أثر

Gambar 3.5 menunjukkan potongan kode program dalam Visual Basic 6.0 untuk melakukan penyesuaian terhadap partikel awalan suatu kata.

Inputan yang akan diproses adalah string kata benda dalam variable 'kata\_arab', sedangkan output kata yang akan dihasilkan adalah pada variable 'arb'. Untuk mendeteksi seluruh huruf pada kata inputan, kata tersebut perlu dibersihkan dulu dari spasi di depan atau di belakangnya. Pembersihan dari spasi ini bisa dilakukan dengan perintah 'Trim'. Variable 'arb' diinisialisasi dulu dengan pemberian string kosong. Selanjutnya dilakukan penyeleksian kondisi, apakah kata-kata tersebut memiliki partikel-partikel awalan : 'ال', 'وال', 'كال', dan 'بال'. Jika terdiri dari awalan-awalan itu, maka awalan-awalan tersebut akan dibersihkan.

Penyesuaian selanjutnya adalah penyesuaian terhadap akhiran suatu kata. Akhiran suatu kata yang diterapkan dalam CAT ini hanyalah akhiran "", karena pengembangan CAT dalam tahap awal ini belum mengakomodasi adanya pelekatan *dhomir* (kata ganti personal). Proses penghilangan awalan dan akhiran ini juga bisa disebut sebagai proses *stemming* dengan mengadopsi konsep *Light Stemming* (Larkey, 2005).

- b. Pencarian di kolom kata jama', jika tidak ditemukan kata yang tepat character per character, maka dilakukan penyesuaian dengan menghilangkan awalan dan akhiran pula. Jika pencarian di tabel Jama tidak mendapatkan kata yang cocok, maka akan dilakukan tahap selanjutnya, tahap 4: tabel Induk1 (List Kata Kerja).

4. Tabel Induk1 (List Kata Kerja).

Data yang tersimpan di tabel Induk1 adalah kata kerja lampau (*fi'il madhi*) saja, dengan huruf-huruf pembentuk *fi'il*. Tabel Induk1 ini adalah sumber bagi proses *tashrif* yang dilakukan dengan *tashrif system*. Tabel 3.13 menunjukkan contoh isian data yang tersimpan dalam tabel Induk1.

Tabel 3.13 Contoh Isian Data di Tabel Induk1 (List Kata Kerja)

non_harakat	Arab	Indonesia	kd_jenis	kd_jml	kd_wazan	fa	ain	lam
حَمَلًا	حَمَلًا	membawa	1	1	4	ح	م	ل
مَرَّ	مَرَّ	Memotong	1	1	1	ب	م	ر
كَبِيرًا	كَبِيرًا	Besar perutnya	1	1	3	ب	ح	ر
أَنْعَمَ	أَنْعَمَ	Memancarkan	1	1	1	ب	ح	م
أَنْعَمَ	أَنْعَمَ	Memotong dengan pedang	1	1	3	ب	ح	ع
أَنْعَمَ	أَنْعَمَ	Gembira	1	1	1	ب	ح	ل
أَنْعَمَ	أَنْعَمَ	Diam karena takut	1	1	1	ب	ح	م
أَنْعَمَ	أَنْعَمَ	Menjadi mumi	1	1	5	ب	ح	م
أَنْعَمَ	أَنْعَمَ	Mempelajari	1	1	3	ب	ح	م

Jika pada proses pencarian di List Kata Kerja tidak ditemui, maka dilakukanlah proses analisis dengan *tashrif system*. Proses analisis dengan *tashrif system* ini akan dijabarkan dalam Bab 4.

```

t1 = Trim(kata_arab)
arb=""
If Mid(t1, 1, 2) = "ل" And Mid(t1, 1, 3) <> "الل" Then
    pan = Len(t1)
    p = pan - 2
    arb = Mid(t1, 3, p)
End If

If Mid(t1, 1, 3) = "وال" And Mid(t1, 1, 4) <> "والل" Then
    pan = Len(t1)
    p = pan - 2
    arb = Mid(t1, 4, p)
End If

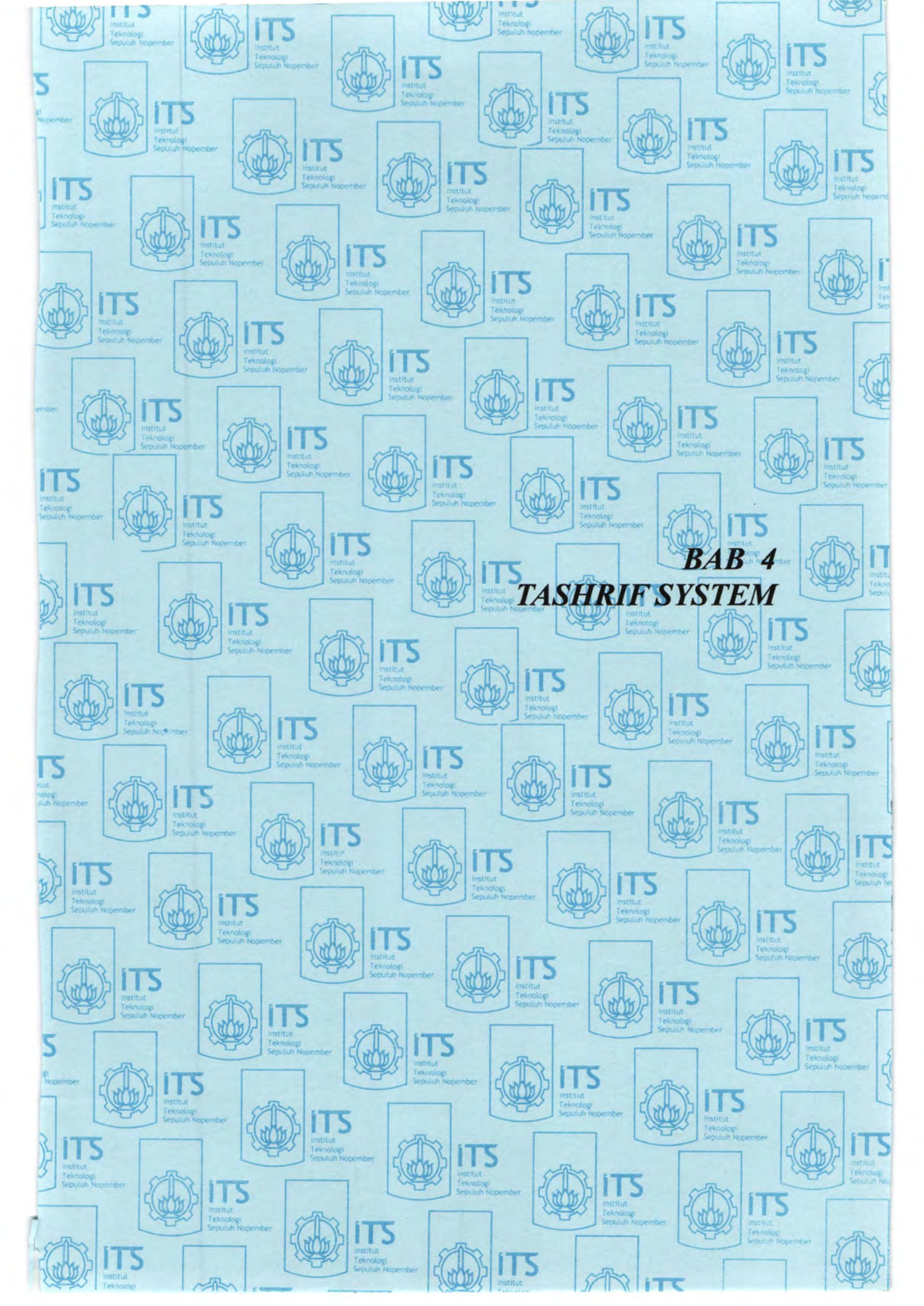
If Mid(t1, 1, 3) = "كل" Then
    pan = Len(t1)
    p = pan - 2
    arb = Mid(t1, 4, p)
End If

If Mid(t1, 1, 3) = "بال" And Mid(t1, 1, 4) <> "بالل" Then
    pan = Len(t1)
    p = pan - 2
    arb = Mid(t1, 4, p)
End If

```

Gambar 3.5 Potongan Kode Visual Basic 6 untuk Penyesuaian  
Partikel Awalan Kata Benda



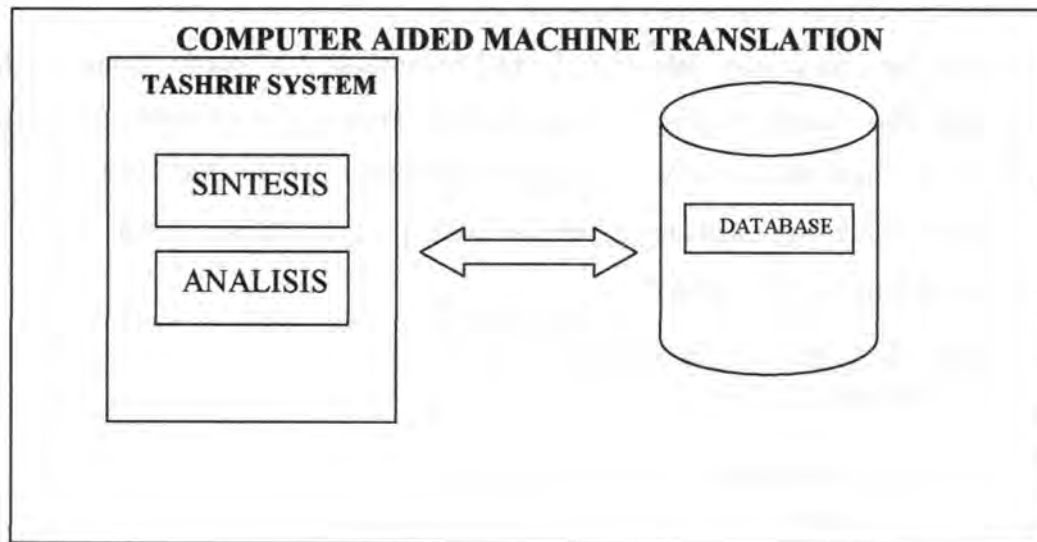


**BAB 4**  
**TASHRIF SYSTEM**

## BAB 4 TASHRIF SYSTEM

### 4.1 Pendahuluan

Sistem tashrif (tashrif system) adalah sebuah modul software yang bisa diintegrasikan dengan sebuah Computer Aided Translation Arab-Indonesia. Ide tashrif system dilatarbelakangi oleh tinjauan terhadap proses penerjemahan yang biasa dilakukan manusia dari bahasa Arab ke bahasa yang lain. Seorang penerjemah dari bahasa Arab biasanya akan menerjemahkan suatu kata atau frase ke dalam bahasa lain dengan menggunakan kamus kata virtual yang ada dalam benaknya. Jika dalam kosakata yang dimilikinya (baik dalam otaknya atau lewat kamus teks) tidak didapati adanya suatu kata tertentu, dia akan cenderung mentashrif kata tersebut untuk mencari padanan kata yang paling sesuai.



Gambar 4.1 Skema Tashrif System

Pada gambar 4.1 terlihat bahwa *Tashrif system* adalah merupakan subsistem dari *Computer Aided Translation* yang dirancang. Jika suatu kata tidak ditemukan secara persis (*exact match*), maka akan dilakukan proses *tashrif*. Dalam *tashrif system* sendiri terdapat 2 mesin utama, yaitu sintesis dan analisis.

Modul analisis akan mengekstrak suatu kata inputan menjadi beberapa kemungkinan kata yang dimaksud. Ekstraksi ini mencakup proses *stemming*, yaitu mengambil partikel-partikel kata di depan atau di belakangnya untuk

mendapatkan huruf-huruf inti. Huruf-huruf inti inilah yang akan dicocokkan lagi dengan database atau disesuaikan dengan *template* yang ada.

Sedangkan modul sintesis akan memproses suatu kata yang telah diketahui kata dasar/ *fi'il madhi*-nya. Dengan *template* yang dimiliki, ia akan menunjukkan pada user kemungkinan kata-kata yang dimaksud. Sebagai contoh, setelah diketahui *fi'il madhi*-nya, ia akan melakukan *tashrif istilahy* menjadi kata-kata berjenis lain, seperti *fi'il mudhari'*, *mashdar*, *mashdar mim*, *fa'il*, *maf'ul*, *fi'il amr*, *fi'il nahi*, *isim zamaan/makan*, dan *isim alat*.

#### 4.2 Subsistem Analisis

Obyek analisis dari proses *tashrif* adalah kata. Jika inputannya adalah frase atau kalimat, inputan itu akan dipecah menjadi kata-kata penyusunnya. Kata-kata yang tidak ditemukan dalam database inilah yang akan dianalisis. Sumber datanya adalah tabel Induk1 (List Kata Kerja).

Kata yang tersimpan dalam tabel Induk1 hanyalah *fi'il madhi*. Namun kata jenis lain bisa dideteksi dengan proses *tashrif*. Proses *tashrif* dalam subsistem analisis ini hanya dibatasi pada kemampuan mendeteksi *fi'il mudhari'* saja.

Gambar 4.2 menunjukkan potongan kode program dalam Visual Basic 6 untuk mendeteksi *fi'il mudhari'*.

```
If Mid(arb, 1, 1) = "ي" Or Mid(arb, 1, 1) = "ت"  
Or Mid(arb, 1, 1) = "ن" Then  
  
    pan = Len(arb)  
    p = pan - 1  
    Ind = Mid(arb, 2, p)  
End if
```

Gambar 4.2 Kode Visual Basic 6 untuk Mendeteksi *Fi'il Mudhari'*

'Arb' adalah variabel untuk inputan kata dalam bahasa Arab. Output kata yang akan dihasilkan disimpan dalam variabel 'Ind'. Jika huruf pertamanya mengandung huruf *dhamir* ي, ت, atau ن, maka itu adalah *fi'il mudhari'*. Perlu dipahami bahwa proses ini adalah setelah melalui tahapan pencocokan pada tabel contoh, *word\_alignment*, *jama*, *induk1*(list kata kerja) seperti dijelaskan dalam

Bab 3. Jika telah terdeteksi kata itu sebagai *fi'il mudhari'*, maka sisa gabungan huruf yang lain, dimulai dari huruf ke 2 sampai terakhir, dicari dalam tabel Induk1 (List Kata Kerja).

#### 4.3 Subsistem Sintesis

Subsistem sintesis adalah kumpulan template yang diadopsi dari literatur kitab *al-Amtsilatut Tashrifiyah* (Maksum bin Ali, 1967) dengan disertai aturan-aturan untuk mengekstrak template tersebut. Penyesuaian terhadap template yang ada di literatur dilakukan dalam beberapa hal:

##### 4.3.1. Penyesuaian Pola

Mayoritas kata dalam bahasa Arab adalah kombinasi dari 3 huruf. Tiga huruf acuan adalah : fa' ( ف ), 'ain ( ع ), dan laam ( ل ). Pola yang ada seperti ditunjukkan pada Gambar 2.1. Untuk penyusunan *template*, contoh-contoh kata dalam literatur referensi yang tidak secara eksplisit mengandung unsur huruf fa', 'ain, dan laam tersebut disubstitusi dengan pola yang sesuai.

##### 4.3.2. Penyesuaian Harakat pada Mashdar dan Mashdar Mim

Dalam referensi literatur yang berorientasi contoh, terlihat perbedaan antara *mashdar mim* dan *isim zamaan/makaan*. Perbedaan ini terjadi karena orientasi peruntukan pembelajaran adalah untuk manusia walaupun sebenarnya kedua jenis kata tersebut memiliki pola yang sama. Tabel 4.1 menunjukkan perbedaan contoh template di literature referensi antara *isim zamaan/makaan* dengan *mashdar mim*.

Pada tabel 4.1 terlihat perbedaan contoh template untuk *isim zamaan* dengan *mashdar miim*. Untuk *fi'il madhi* *نَصَرَ* *mashdar mimnya* adalah *مَنْصَرًا* sedangkan *isim zamaannya* adalah *مَنْصَرٌ*, perbedaan terlihat pada harakat terakhir. *Mashdar miim* berharakat *fathatain*, sedangkan *isim zamaan* berharakat *dhommatain*.

Tabel 4.1 Perbedaan Contoh Template pada Literatur antara *Mashdar Mim* dengan *Isim Zamaan*

Isim Zamaan	Masdar Mim	Fi'il Madhi	Wazan
مَنْصَرٌ	مَنْصَرًا	نَصَرَ	فَعَلَ - يَفْعُلُ
مَفْتَحٌ	مَفْتَحًا	فَتَحَ	فَعَلَ - يَفْعَلُ
مَعْلَمٌ	مَعْلَمًا	عَلِمَ	فَعَلَ - يَفْعَلُ
مَحْسَنٌ	مَحْسَنًا	حَسَنَ	فَعَلَ - يَفْعَلُ

Pada tashrif system, isim *zamaan/ makaan* dengan *mashdar miim* dibuatkan pola yang sama. Sehingga penempatan pola kata pada masing-masing bagian menjadi seperti ditunjukkan dalam Tabel 4.2.

Tabel 4.2 Peletakan Template Berbasis Pola pada Tashrif System

JUNLAH HURUF:      النامى اشرد      3 HURUF ASLI.										
WAZAN:      فعُل - يفعُلُ										
اسم الألة	اسم زمان و مكان	فعل ناهي	فعل أمر	مفعول	فاعل	مصدر ميم	مصدر	فعل مضارع	فعل ماض	
منعُ	منعُ	لا منعُ	انمعُ	منعونُ	فانعُ	منعلا	فعلا	نمنعُ	منع	صحيح
منعُ	منعُ	لا منعُ	فمنعُ	منعونُ	فانعُ	منعُ	فمنعُ	نمنعُ	منع	مضاعف
منعونُ	منعانُ	لا منعُ	فمنعُ	منعونُ	فانعُ	منعانُ	فمنعونُ	نمنعونُ	منعون	أجوف راوي فان
منعُ	منعُ	لا منعُ	انمعُ	منعونُ	فانعُ	منعُ	فمنعونُ	نمنعونُ	منعون	ناقص راوي يعي
منعُ	مناعُ	لا ناعُ	انراعُ	مناعونُ	انراعُ	مناعُ	فمناعونُ	نمنعونُ	منعون	مهوز فاء

Pada Tabel 4.2 terlihat pada baris مهموز فاء , ناقص واوي , أجوف واوي , مضاعف ,  
 terdapat kesamaan pola pada kolom اسم زمان و مكان مصدر ميم .

### 4.3.3. Pengubahan Orientasi dari Kolom ke Baris

Untuk memudahkan proses pencarian (*searching*) dan menepatkan pola yang sesuai (*matching*), peletakan pola tersebut di atas disusun dalam orientasi baris. Untuk jumlah huruf : 3 huruf asli, wazan فَعْلٌ - يَفْعُلٌ , dan bina' shohih, pola pada tashrif system ditunjukkan dalam Tabel 4.3.

Tabel 4.3 Pengubahan Orientasi Baris dan Kolom

Wazan	Jenis	Tashrif	Template
فَعْلٌ - يَفْعُلٌ	صحيح	فعل ماض	فعل
فَعْلٌ - يَفْعُلٌ	صحيح	فعل مضارع	يَفْعُلٌ
فَعْلٌ - يَفْعُلٌ	صحيح	مصدر	فَعْلٌ
فَعْلٌ - يَفْعُلٌ	صحيح	مصدر ميم	مَفْعُلٌ
فَعْلٌ - يَفْعُلٌ	صحيح	فاعل	فَاعِلٌ
فَعْلٌ - يَفْعُلٌ	صحيح	مفعول	مَفْعُولٌ
فَعْلٌ - يَفْعُلٌ	صحيح	فعل أمر	اَفْعُلْ
فَعْلٌ - يَفْعُلٌ	صحيح	فعل ناهي	لا تَفْعُلْ
فَعْلٌ - يَفْعُلٌ	صحيح	اسم زمان	مَفْعُلٌ
فَعْلٌ - يَفْعُلٌ	صحيح	اسم الآلة	مَفْعُلٌ

Pada Tabel 4.2 Jenis – jenis tashrif dibuat dalam orientasi baris, sedangkan di Tabel 4.3 orientasinya adalah kolom. Demikian juga untuk jenis bangunan kata, wazan dan template.

Karena pengolahan software akan memanfaatkan *Database Management System* (DBMS), maka pengaturannya juga disesuaikan dengan mekanisme penyimpanan dalam DBMS. Untuk menghemat ruang penyimpanan dan mempercepat proses pencarian, masing-masing kolom wazan, jenis, dan tashrif-

nya cukup diisi dengan nomor kode-nya. Sehingga didapatkan mekanisme peletakan pola kata seperti pada Tabel 4.4.

Pada Tabel 4.3 wazan, jenis, dan tashrif direpresentasikan dalam bentuk string, sedangkan pada Tabel 4.4 dalam bentuk kode. Hal ini dilakukan untuk efisiensi penyimpanan data dalam database. Demikianlah mekanisme penyusunan template pada *Tashrif System*. Tabel pada database yang menyimpan data template tersebut terdiri dari 1530 record.

Saat ini, *tashrif system* yang dirancang hanya untuk mengakomodasi tashrif istilah. Rencana pengembangan selanjutnya, sistem ini bisa juga mengakomodasi *tashrif lughowy*, yaitu analisis morfologi kata berdasarkan kata ganti (*dhomir*). Jika pengembangan ini dilakukan, perlu adanya modifikasi penyediaan template dengan penambahan field jenis kata ganti. Pengembangan tersebut berakibat pada modifikasi susunan tabel template dengan tambahan field *kd\_dhamir*, sehingga menjadi seperti ditunjukkan Tabel 4.5.

Tabel 4.4 Peletakan Pola Kata dengan Kode

kd_wazan	kd_jenis	kd_tashrif	Template
1	1	1	فَعْلٌ
1	1	2	نَفْعٌ
1	1	3	فَعْلٌ
1	1	4	مَفْعَلٌ
1	1	5	فَاعِلٌ
1	1	6	مَفْعُولٌ
1	1	7	أَفْعَلٌ
1	1	8	لَا نَفْعٌ
1	1	9	مَفْعَلٌ
1	1	10	مَفْعَلٌ

Template-template yang telah disimpan dalam database bisa dimanfaatkan untuk mencari jenis kata tertentu jika telah diketahui *fi'il madhi*-nya. Jika *fi'il madhi* tersebut terdata dalam tabel Induk1 (List Kata Kerja), maka jenis kata yang

diinginkan bisa didapatkan dengan cara mensubstitusi *fa'*, *ain*, dan *lam fi'il* dengan kata yang terdapat pada template.

Tabel 4.5 Peletakan Pola Template dengan Tambahan Kolom Kd\_Dhamir

kd_wazan	kd_jenis	kd_tashrif	kd_dhamir	template
1	1	1	1	فَعَلَ
1	1	1	2	فَعَلَا
1	1	1	3	فَعَلُوا
1	1	1	4	فَعَلْتُ
1	1	1	5	فَعَلْنَا
1	1	1	6	فَعَلْنِ
1	1	1	7	فَعَلْتِ
1	1	1	8	فَعَلْتُمَا
1	1	1	9	فَعَلْتُمْ
1	1	1	10	فَعَلْتِ
1	1	1	11	فَعَلْتُمَا
1	1	1	12	فَعَلْتُنَّ
1	1	1	13	فَعَلْتُ
1	1	1	14	فَعَلْنَا





**BAB 5**  
**UJI COBA DAN PENGEMBANGAN**  
**LEBIH LANJUT**

## BAB 5

### UJI COBA DAN PENGEMBANGAN LEBIH LANJUT

Pada bab ini akan dijelaskan tentang ujicoba terhadap prototipe *Computer Aided Translation Arab-Indonesia* yang telah dikembangkan berikut hasilnya. Didahului oleh uraian tentang data yang digunakan, *training set* dan *testing set*, serta skenario percobaan. Hasil ujicoba dilaporkan secara kuantitatif dengan pengukuran terhadap presisi. Di akhir bab akan dipaparkan tentang kesimpulan hasil ujicoba serta saran-saran untuk pengembangan lebih lanjut.

#### 5.1 Data

##### 5.1.1 Data Inputan

Data yang diujicobakan adalah kata atau kalimat dalam bahasa Arab tanpa harakat, bisa terdiri dari kata benda maupun kata kerja. Kata kerja yang diisikan bisa berupa kata kerja lampau (*fi'il madhi*) atau kata kerja sekarang dan akan datang (*fi'il mudhari'*). *Fi'il madhi* yang bisa diisikan adalah untuk kata ganti personal ke-3 laki-laki, sedangkan *fi'il mudhari'* mencakup seluruh kata ganti personal dalam bentuk aktif, dan tidak terhubung langsung dengan obyek penderita. Kata benda yang diinputkan bisa berupa *nakirah* (tanpa imbuhan ' ال ') maupun *ma'rifah* (dengan imbuhan ' ال '), baik tunggal ataupun jama' (lebih dari 2).

##### 5.1.2 Training Data

Training data adalah data kata atau kalimat yang tercakup dalam database sistem. Cakupan kata/ kalimat tersebut terdiri dari :

1. Contoh terjemahan, terdiri dari 15 kalimat contoh.

Contoh terjemahan diambilkan dari beberapa jenis :

- a. Contoh – contoh kalimat dalam buku-buku pelajaran bahasa Arab tentang pengenalan bentuk *mudhaf – mudhaf ilaih, muftada' – khabar, jar – majruur*. Kalimat-kalimat sederhana yang masuk dalam kriteria ini di antaranya:

ذهب حامد إلى المدرسة

Artinya: Hamid pergi ke sekolah

ينام عباس فوق السرير

Artinya: Abbas sedang tidur di atas tempat tidur

تسقط الورقة تحت الشجرة

Artinya: Daun jatuh dari pohon

كتب المدرس الدرس على السبورة

Artinya: Guru menulis pelajaran di papan tulis

b. Ungkapan – ungkapan lafadz *dzikir*, seperti:

استغفر الله العظيم

Artinya: Aku memohon ampunan kepada Allah Yang Maha Agung

إنا لله وإنا إليه راجعون

Artinya: Sesungguhnya kami milik Allah dan sesungguhnya kepadaNya kami kembali

c. Potongan kalimat yang terdapat dalam AlQur'an, misalkan:

إن الله هو الرزاق ذو القوة المتين

Artinya: Sesungguhnya Allah adalah Yang Maha Pemberi Rezeki Yang Memiliki Kekuatan yang teguh.

لله ما في السماوات وما في الأرض

Artinya: Milik Allah sajalah segala yang ada di langit dan di bumi.

d. Potongan kalimat yang terdapat dalam naskah teks Fiqh Islam, di antaranya:

فكل الماء طهور ما لم تخالطه النجاسة

Artinya: Maka semua air adalah suci selama tidak tercampuri najis

الماء طهور لا ينجسه شيء إلا ما غير طعمه أو ريحه

Artinya: Air suci tidak ternajiskan dengan sesuatu kecuali jika berubah rasa atau baunya.

2. *Word Alignment*, kata – kata penyusun kalimat contoh yang sudah disejajarkan dengan padanan kata bahasa Indonesianya. Dalam database terdata 92 records.
3. Kamus kata benda, terdiri dari field kata benda tunggal dan kata benda jama'. Dalam database terdata 486 records.
4. List Kata kerja, berisikan kata kerja lampau untuk kata ganti orang ke-3 berjenis laki-laki, sebanyak 6826 records.

### 5.1.3 Testing Data (Data Tes)

Data tes yang digunakan dalam pengujian adalah :

1. *E-book* berbahasa Arab berjudul : ' الملخص الفقه ' yang disusun Syaikh Shalih al-Fauzan. Buku tersebut berisi penjelasan ringkas tentang Fiqh dalam Ilmu Agama Islam. Sampel data yang diambil adalah dari bab tentang kesucian air.
2. Halaman awal portal berita online al-jazeera dalam bahasa Arab yang di-*download* pada tanggal 15 Agustus 2008.

## 5.2 Skenario Percobaan

Pengujian terhadap kemampuan *prototype* yang dikembangkan dilakukan untuk mengetahui reaksi penerjemahan terhadap inputan dalam bentuk kata, frase, atau kalimat. Percobaan dilakukan terhadap *training data* dan *testing data*.

### 5.2.1 Skenario I : Percobaan terhadap *Training Data*

Percobaan dirancang untuk melihat kemampuan system dalam hal :

1. Menghasilkan output terjemahan bahasa Indonesia dari inputan satu kata berbahasa Arab.

Untuk menguji kemampuan system tersebut dilakukan inputan satu kata bahasa Arab yang tidak berharakat dengan tiga kemungkinan :

- a. Kata tersebut ada dalam tabel kamus kata.
  - b. Kata tersebut tidak ada dalam tabel kamus kata namun memiliki akar kata yang sama dengan suatu kata kerja dalam tabel list kata kerja.
  - c. Kata tersebut tidak terdapat dalam tabel kamus kata dan juga tidak memiliki akar kata yang sama dengan seluruh kata dalam tabel list kata kerja, namun terdapat dalam tabel sinonim kata.
2. Mengakomodasi inputan frase atau kalimat dalam bahasa Arab untuk dikonversi menjadi padanannya dalam bahasa Indonesia.

Sistem akan diuji untuk melakukan reaksi terhadap inputan dengan kriteria sebagai berikut :

- a. Kalimat atau frase tersebut ada dalam tabel contoh kalimat terjemahan (*Example Base*) secara persis sama karakter demi karakter.
- b. Sebagian kalimat atau frase tersebut ada dalam salah satu baris contoh kalimat terjemahan, sebagian lain ada dalam baris yang lain, sedangkan sebagian kata ada yang terdapat pada baris (*row*) lain di tabel contoh kalimat terjemahan.
- c. Seluruh bagian kalimat atau frase inputan tidak ada yang terdapat dalam tabel contoh kalimat terjemahan, namun tiap-tiap katanya ada dalam tabel kamus kata.

### **5.2.2 Skenario II : Percobaan terhadap *Testing Data***

Percobaan terhadap *testing data* dilakukan dengan memasukkan frase/kalimat yang ada pada *testing data* secara random. Frase atau kalimat tersebut diinputkan pada sistem untuk melihat keakuratan terjemahan yang dihasilkan.

## **5.3 Pengujian terhadap Training Data**

### **5.3.1 Pengujian 1 Kata**

Dilakukan 20 kali ujicoba pengentryan pada system dengan hasil seperti pada tabel 5.1.

Pengujian satu kata pada training data ini dilakukan dengan beberapa kriteria :

- a. Pengujian kata kerja : no 1 sampai dengan 10. Semuanya adalah *fi'il madhi*, kecuali pengujian no 5 adalah *fi'il mudhari'* . Hasil yang didapatkan dari pengujian no 5 adalah dari tashrif system, ekstraksi langsung dari database.
- b. Pengujian kata benda dilakukan dari no 11 sampai dengan 20. Pengujian no 11 adalah untuk kata benda *ma'rifah* dengan penanganan 'ال' imbuhan', sedangkan 12 sampai 20 adalah kata benda *nakirah*.

Tabel 5.1 Ujicoba pengentryan satu kata pada training data

NO	INPUT	OUTPUT
1	رضي	Rela
2	سمح	murah/ dermawan
3	ينع	Matang, Menjadi sangat merah
4	منح	Memberi
5	يمنع	sedang Mencegah
6	شمال	Berhembus dari Utara
7	متع	Panjang, Naik, Membawa pergi, Kuat, Mendustakan
8	صدق	Benar
9	رغب	Ingin, Berkehendak
10	حسن	Bagus, Baik, cantik
11	الكلب	Anjing
12	ورقة	Daun
13	بيت	Rumah
14	تلميذ	Murid
15	صغير	Kecil
16	جند	Tentara
17	ضريير	Buta
18	سوق	Pasar
19	ضامن	Penanggungjawab
20	رديء	buruk; jelek

### 5.3.2 Pengujian Kalimat atau Frase

Uji coba dilakukan 20 kali terhadap frase / kalimat yang tercakup dalam database sistem, baik yang terdapat pada potongan contoh terjemahan, kamus kata benda, kata kerja, ataupun diproses dengan tashrif system. Input kata yang dimasukkan beserta output yang dihasilkan tampak pada Tabel 5.2.

Percobaan no 1 pada frase : ' الحمد لله رب العالمين ' menghasilkan output kalimat terjemahan : ' pujian bagi Allah Tuhan semesta alam ' . Hasil tersebut didapatkan dari beberapa sumber:

- (a). Kata ' الحمد ' sumbernya adalah kamus kata benda, yang terdapat di dalamnya kata 'حمد' dengan penyesuaian algoritma yang mampu menangani penambahan partikel 'ال'.
- (b) Kata 'الله' diambil dari potongan contoh terjemahan 'السموات والأرض' yang berarti 'untuk Allah'.
- (c) Kata 'رب' sumbernya adalah kamus kata benda dengan kesamaan persis karakter demi karakter yang berarti 'Tuhan'.
- (d) Kata 'العالمين' diambil dari kamus kata benda dengan penyesuaian partikel 'ال'.

Percobaan nomor 2 didapatkan dari suatu contoh terjemahan yang persis sama kata demi kata dan sesuai urutannya dari awal sampai akhir. Percobaan nomor 3 menunjukkan keberhasilan ekstraksi dari 2 contoh terjemahan yang berbeda. Kalimat ' استغفر الله ' ada dalam satu contoh, sedangkan kata ' الغفار ' diekstraksi dari kalimat lain yaitu ' الله الغفار ' .

Kalimat ' حامد يقرأ الكتب ' diinputkan pada percobaan nomor 4. Kata حامد terdapat dalam tabel Word\_alignment. Kata kerja يقرأ adalah pengembangan dengan *tashrif system* untuk kata yang ada dalam List Kata Kerja ' قرأ ' yang artinya membaca. Karena terjadi perubahan dari *fi'il madhi* menjadi *fi'il mudhari'*, maka ada tambahan kata terjemahan 'sedang', sehingga output terjemahan yang dihasilkan dari kata kerja ini adalah: 'sedang membaca'. Kata ' الكتب ' tidak ada dalam seluruh cakupan, namun merupakan pengembangan dari kata yang terdaftar dalam tabel Jama yang mampu mengakomodasi imbuhan 'ال'. Kata yang terdapat dalam tabel Jama adalah كتب yang merupakan bentuk jama' dari kata 'buku/kitab'.

Percobaan nomor 5 pada kalimat: 'سكن حامد المسجد' menghasilkan output terjemahan 'Tinggal, Menetap Hamid masjid'. Potongan kata-kata tersebut terdapat pada beberapa sumber :

- (a) سكن terdapat dalam tabel Induk1/ List Kata Kerja yang artinya adalah 'Tinggal/ menetap'.
- (b) حامد terdapat dalam tabel Word\_alignment yang artinya: 'Hamid'.
- (c) المسجد pengembangan dari kata مسجد yang terdapat dalam daftar kata benda tunggal dengan kemampuan mengenali partikel 'ال'.

Percobaan nomor 7 menghasilkan output kata yang tidak sepenuhnya tepat. Kalimat yang diinputkan adalah: كتب حامد الكتاب yang seharusnya menghasilkan output: 'menulis Hamid kitab', menjadi buku/ kitab Hamid buku/ kitab. Hal ini terjadi karena kata كتب terdapat dalam tabel kata benda *jama'* dan juga terdapat pada List Kata Kerja, sedangkan *system* langsung memilih yang ada pada kata benda *jama'*nya.

Percobaan nomor 8 dilakukan dengan menginputkan kata : هذا كتاب كبير .

Semua kata-kata tersebut: هذا, كتاب, dan كبير terdapat dalam daftar kata benda.

Demikian juga dengan percobaan nomor 9, semua kata-kata penyusunnya tercover dalam daftar kata benda.

## 5.4 Pengujian terhadap *Testing Data*

### 5.4.1 Pengujian terhadap Data Tes pada Domain Ilmu Fiqh (الملخص الفقهي)

Tabel 5.3 menunjukkan daftar inputan kata yang dimasukkan dalam system serta output yang dihasilkan. Pengujian dilakukan dengan mengambil frase dalam e-book tersebut dan dimasukkan sebagai input sistem dalam bahasa Arab. Jika ditemukan padanan katanya, output akan menampilkan kata yang sesuai dalam bahasa Indonesia, sedangkan jika tidak, maka akan ditampilkan string '[]'. Sehingga, string '[]' menunjukkan bahwa ada bagian dari kata yang tidak tercover dalam sistem. Jika user ingin petunjuk lebih lanjut, sistem akan menampilkan pensejajaran kata demi kata dari kalimat tersebut dengan menekan tombol 'PETUNJUK' seperti dalam Gambar 5.1



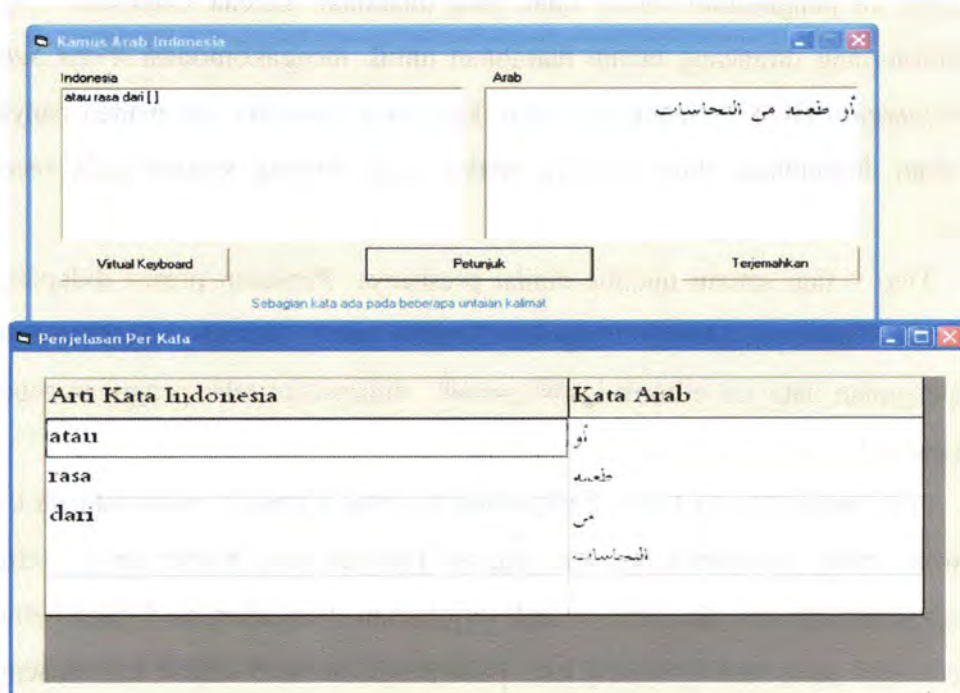
Tampak pada gambar 5.1 bahwa setelah ditampilkan petunjuk, kata 'النجاسات' tidak memiliki padanan kata dalam bahasa Indonesia karena tidak tercover dalam system.

Tabel 5.2 Pengujian Inputan Kalimat/ Frase pada Training Data

NO	INPUT	OUTPUT
1	الحمد لله رب العالمين	pujian bagi Allah Tuhan semesta alam
2	استغفر الله العظيم	aku meminta ampun kepada Allah Yang Maha Agung
3	استغفر الله الغفار	aku meminta ampun Allah Yang Maha Pengampun
4	حامد يقرأ الكتب	Hamid sedang Membaca buku/ kitab
5	حامد يقرأ القرآن	Hamid sedang Membaca Al-Qur'an
6	سكن حامد المسجد	Tinggal, Menetap Hamid masjid
7	كتب حامد الكتاب	buku/ kitab Hamid buku/ kitab
8	هذا كتاب كبير	ini buku/ kitab besar
9	هذا بيت صغير	ini rumah kecil
10	لعب الأطفال أمام المسجد	Bermain anak kecil (j) di depan masjid
11	هو يذهب إلى السوق	dialah sedang Pergi ke pasar
12	هو يأخذ الورقة من الشجرة	dialah sedang Mengambil lembaran dari pohon
13	سمع حامد الصوت	Mendengar Hamid suara
14	صدق رسول الله	Benar utusan Allah
15	المسجد طهور لا ينجسه	Masjid suci tidak temajiskan
16	يدرس عباس اللغة العربية	sedang Belajar Abbas Bahasa Arab
17	تسقط إبرة من المكتب	jatuh jarum dari meja tulis
18	يلبس عباس النظارة	sedang Memakai Abbas kaca mata
19	إننا نركب الحصان	Sesungguhnya kami sedang Mengendarai kuda
20	حامد حسن الوجه	Hamid Bagus, Baik, cantik Dihormati,Dimuliakan wajah

Tabel 5.3 Pengujian terhadap Data Tes pada Domain Ilmu Fiqh

NO	INPUT	OUTPUT
1	والماء طاهر مطهر لا يخرج عن الوصفين	dan air(jama') suci mensucikan tidak mengeluarkannya dari dua sifat
2	إلا ما غير ربه أو لونه	kecuali selama berubah baunya atau warnanya
3	أو طعمه من النجاسات	atau rasa dari [ ]
4	ما أخرج عن اسم الماء المطلق	selama [ ] dari [ ] air [ ]
5	من المنخيرات الطاهرة	dari [ ]
6	ولا فرق بين قليل وكثير	[ ] Menceraikan antara sedikit [ ] banyak
7	وما فوق الظنن وما دونهما	dan segala di atas [ ] dan segala [ ]
8	ومحرك ساكن ومستعمل وغير مستعمل	[ ] bergerak diam [ ] bekas dipakai [ ] berubah bekas dipakai
9	من بئر بضاعة	dari sumur bidho-ah
10	وهي بئر تلقى فيها الحوض وأحرم الكلاب	[ ] sumur [ ] padanya darah haid [ ] anjing



Gambar 5.1 Screenshot tampilan Petunjuk Kata per Kata

#### 5.4.2 Pengujian terhadap Data Tes pada Situs Portal Berita Al-Jazeera

Tabel 5.4 Pengujian terhadap Data Tes pada Situs Berita

NO	INPUT	OUTPUT
1	قتلى بسلسلة هجمات انتحارية نفذتها نساء بالعراق	bangkai; jenazah [ ] [ ] [ ] [ ] wanita [ ]
2	لقي 26 شخصا على الأقل مصرعهم بهجمات	Bertemu [ ] [ ] pada [ ] [ ] [ ]
3	حزن بريطاني على رحيل درويش	Sedih [ ] pada [ ] [ ]
4	كاركاتير: رمضان	[ ] ramadhan

Dari pengujian terhadap data tes pada situs berita nampak bahwa lebih banyak kata yang tidak tercover oleh system, terlihat dari banyaknya string '['] ditampilkan sebagai output.

### 5.5 Analisis dan Evaluasi

Penilaian terhadap hasil ujicoba ini menggunakan pengukuran presisi. Pengukuran presisi dipilih karena hal yang dicari adalah ketepatan kata demi kata antara input dengan output. Biasanya dalam sebuah *Information Retrieval* pengukuran presisi (*precision*) diikuti dengan pengukuran *recall*. Namun, dalam percobaan ini pengukuran *recall* tidak bisa dilakukan karena Computer Aided Translation yang dirancang belum diarahkan untuk mengakomodasi *Word Sense Disambiguation (WSD)*. Sehingga, untuk kata yang memiliki terjemahan banyak, yang akan ditampilkan dulu hasilnya adalah yang ditemui system pada record teratas.

Tiap – tiap satuan ujicoba dinilai presisinya. Penilaian presisi didapatkan dari jumlah output yang benar dibagi keseluruhan jumlah output yang diharapkan. Pada pengujian data tes e-book 'الملخص الفقهي' didapatkan nilai pengujian seperti pada tabel 5.5.

Pada pengujian no 1 dan 2 presisinya bernilai 1 karena untuk seluruh kata inputan memiliki padanan dalam kata bahasa Indonesianya. Kesamaan ini terjadi karena kalimat inputan ada pada contoh terjemahan. Pengujian no 9 juga bernilai 1. Kecocokan pada masing-masing kata pada pengujian no 9 adalah karena semua kata pembentuknya terdapat dalam kamus kata benda.

Tabel 5.5 Nilai Pengujian terhadap Data Tes pada 'الملخص الفقهي'

NO	INPUT	OUTPUT	PRESISI
1	والماء طاهر مطهر لا يخرج عن الوصفين	dan air(jama) suci mensucikan tidak mengeluarkannya dari dua sifat	1
2	إلا ما غير ربحه أو لونه	kecuali selama berubah baunya atau warnanya	1
3	أو طعمه من الأجاسات	atau rasa dari [ ]	0,75
4	ما أخرج عن اسم الماء المطلق	selama [ ] dari [ ] air [ ]	0,5
5	من المغيرات الطاهرة	dari [ ]	0,333333333
6	ولا فرق بين طين وكثير	[ ] Menceraikan antara sedikit [ ] banyak	0,583333333
7	وما فوق الثنتين وما دونهما	dan segala di atas [ ] dan segala [ ]	0,714285714
8	ومتحرك ساكن ومستهمل وغير مستعمل	[ ] bergerak diam [ ] bekas dipakai [ ] berubah bekas dipakai	0,625
9	من بئر بضاعة	dari sumur bidho-ah	1
10	وهي بئر يلقى فيها الحيض ولحوم الكلاب	[ ] sumur [ ] padanya darah haid [ ] anjing	0,625
			0,713095238

Pengujian no 3 tidak semuanya memiliki padanan kata. Kata 'النجات' tidak memiliki padanan kata yang tercover pada system. Dari total 4 kata yang diharapkan, kata yang sesuai adalah 3, sehingga nilai presisinya adalah:  $(4-3)/4=3/4=0,75$ .

Pada pengujian no 4 kata 'أخرجه', 'اسم', dan 'المطلق' tidak memiliki padanan kata yang tercover dalam sistem. Jumlah total kesamaan kata yang diharapkan adalah 6, sedangkan kata yang sama adalah 3, sehingga presisinya adalah 0,5.

Presisi terkecil didapat pada pengujian no 5 karena kata yang ditemukan hanyalah 'من' saja, sedangkan kata 'المغيرات' dan 'الطاهره' tidak tercover dalam system. Nilai presisinya adalah 0,333. Setelah dilakukan 10 kali ujicoba, didapatkan total rata-rata presisi keseluruhan adalah 0,71 atau 71%.

Pada pengujian data tes yang kedua, yaitu terhadap portal berita online Al-Jazeera, didapatkan nilai presisi yang lebih rendah, yaitu total rata-rata 0,42 atau 42%. Rendahnya nilai presisi ini terjadi karena banyaknya kata berupa identitas diri seperti nama orang, nama tempat, dan nama organisasi yang tidak tercover dalam sistem.

Tabel 5.6 Hasil Pengujian terhadap Situs Berita Online Al-Jazeera

NO	INPUT	OUTPUT	PRESISI
1	فتى بسنة حساب انحرابه بقديها ساء بالعراف	bangkai; jenazah [ ] [ ] [ ] [ ] [ ] wanita [ ]	0,428571
2	لقي 26 شخصا على الأقل مصرعهم بهجمات	Bertemu [ ] [ ] pada [ ] [ ] [ ] [ ]	0,285714
3	حزن بريطاني على رحيل نرويس	Sedih [ ] pada [ ] [ ] [ ]	0,5
4	كاريناس: رمضان	[ ] ramadhan	0,5
			0,428571

## 5.6. Kesimpulan

Dari pengujian tersebut dapat ditarik beberapa kesimpulan :

1. Pembuatan *Computer Aided Translation* Arab –Indonesia bisa ditopang dengan mesin *Example Based Machine Translation* sebagai referensi pencarian kata yang terhubung dalam suatu konteks dengan kata lain dalam satu kalimat.
2. Dalam penyusunan kamus kata pada suatu *Computer Aided Translation Arab – Indonesia*, tabel kata benda perlu dipisah dari tabel kata kerja,

karena penanganan untuk 2 jenis kata tersebut berbeda. Kata benda bisa mengalami tambahan partikel ' ال ' sedangkan pada kata kerja tidak. Sistem tashrif (*system tashrif*) hanya bisa diterapkan pada kata kerja dengan pengelompokan klasifikasi jenis, jumlah huruf pembentuk, dan jenis wazannya.

3. Sistem tashrif (*tashrif system*) yang berupa kumpulan template bisa diekstraksi untuk mengakomodasi kata yang tidak terdapat dalam kamus kata ataupun contoh terjemahan, namun memiliki akar kata yang sama dengan yang terdapat pada list kata kerja.
4. Kemampuan sistem bisa ditambah dengan penambahan data pada contoh terjemahan, kamus kata benda, maupun pada list kata kerja.

### 5.7 Pengembangan Lebih Lanjut

Selanjutnya, sistem dapat dikembangkan dengan kemampuan sebagai berikut:

1. Mengakomodasi perubahan kata ganti pada setiap kata kerja inputan.  
Software yang dikembangkan masih berasaskan metode tashrif istilah saja, belum mencakup tashrif *lughowy*. Untuk penyesuaian ke arah itu perlu modifikasi pada struktur template dan isian tambahan pada template untuk kata ganti yang sesuai.
2. Mampu berjalan pada multi platform sistem operasi.  
Sementara ini, software yang dirancang adalah *desktop-based*, berjalan pada sistem operasi Microsoft Windows XP. Ke depan, diharapkan ada pengembangan sistem untuk bisa berjalan pada sistem operasi lain semacam Linux dan semisalnya, juga dalam lingkungan *web-based*.
3. Mengakomodasi *Word Sense Disambiguation* (WSD) yaitu pemilihan kata paling tepat jika didapati pada contoh terjemahan beberapa kata yang memiliki kemungkinan penerjemahan yang banyak serta berada pada banyak cakupan (contoh terjemahan, list kata kerja, kamus kata benda).



**DAFTAR PUSTAKA**

## Daftar Pustaka

- Al-Wasiti, Salman(2005). *Al-tafaa'ul baynal insaan wa al-aalah* ( Interaksi antara manusia dengan mesin). Didownload dari <www.almughtarib.com>
- Azhar bin Muhammad(2007) . *Beberapa Aspek Keunikan dan Keistimewaan Bahasa Arab sebagai Bahasa Al-Qur'an*. Universiti Teknologi Malaysia.
- Chen, Aitao dan Grey, Fredic(2002). *Building an Arabic Stemmer for Information Retrieval*. University of California at Berkeley.
- Darwish, Kareem(2002). *Building a Shallow Arabic Morphological Analyzer in One Day*.Electrical and Computer Engineering Dept. University of Maryland, College Park.
- Ethnologue(2005).*Ethnologue:Languages of the Worlds*.15 th ed.
- Hutchin, John(2005).*Towards a Definition of Example-Based Machine Translation*. 89 ChristCurchRoad Norwich NR2 3NG.
- K. Bar, Y. Choueka dan N. Dershowitz(2007). *An Arabic to English Example-Based Translation System*. Tel Aviv University Dept of Computer Science.
- Larkey, Leah S(2005). *Light Stemming for Arabic Information Retrieval*. Univ of Massachusetts.
- Maksum bin Ali (1965). *Al-Amsilatut Tashrifiiyah*. Maktabah Asy-Syaikh Saalim bin Sa'ad Nabhaan.
- Muhammad, Maulana Ibrahim (2006). *From Treasures of Arabic Morphology*. Academy of Islamic Research Madrasah In' amiyah.
- Umam, Chatibul(2004). *Kaidah Tata Bahasa Arab*. Darul Ulum Press.
- William J. Black dan Sabri El-Kateb(2004). *A Prototype English-Arabic Dictionary Based on WordNet*.

## TENTANG PENULIS

Dilahirkan sebagai anak bungsu dari 3 bersaudara di Kraksaan Probolinggo pada 20 Oktober 1979, penulis mengenyam pendidikan dasar di SDN 1 Patokan Kraksaan. Pendidikan menengah yang penuh 'warna' diawali di SMPN 1 Kraksaan dan dituntaskan selama 3 tahun di kota Malang di SMUN 3 Malang. Setelah sempat merasakan 3 semester awal di Fakultas Teknik Jurusan Teknik Elektro Universitas Brawijaya Malang pada 1998-1999 penulis berpindah study pada Sekolah Tinggi Teknologi Nurul Jadid Paiton Probolinggo jurusan Teknik Informatika. Gelar Sarjana Komputer diraih pada tahun 2004 dan setahun setelahnya mendapatkan Beasiswa Program Pasca Sarjana (BPPS) dari Dirjen Pendidikan Tinggi Departemen Pendidikan Nasional RI untuk melanjutkan study di S2 Teknik Informatika ITS Surabaya. Penulis dapat dihubungi di [kharisman08@gmail.com](mailto:kharisman08@gmail.com).