

repository.ub.ac.id

**APLIKASI PENERJEMAH BAHASA INGGRIS KE BAHASA  
INDONESIA MENGGUNAKAN NATURAL LANGUAGE PROCESSING  
(NLP)**

**SKRIPSI**

*Diajukan untuk memenuhi persyaratan  
memperoleh gelar Sarjana Teknik*



Disusun Oleh:  
**AGUNG NUGROHO PRAMUDHITA**  
NIM. 0710630064-63

**KEMENTERIAN PENDIDIKAN NASIONAL  
UNIVERSITAS BRAWIJAYA  
FAKULTAS TEKNIK  
MALANG  
2012**

**LEMBAR PERSETUJUAN**

**APLIKASI PENERJEMAH BAHASA INGGRIS KE BAHASA  
INDONESIA MENGGUNAKAN NATURAL LANGUAGE PROCESSING  
(NLP)**

**SKRIPSI**

**JURUSAN TEKNIK ELEKTRO**

Diajukan untuk memenuhi persyaratan  
memperoleh gelar Sarjana Teknik



Oleh:

**AGUNG NUGROHO PRAMUDHITA**  
**NIM. 0710630064-63**

**Telah mengetahui dan disetujui oleh:**

**Dosen Pembimbing I,**

**Dosen Pembimbing II,**

**Ir. Muhammad Aswin, MT.**  
**NIP. 19650626 19902 1 001**

**Adharul Muttaqin, ST., MT.**  
**NIP. 19760121 200501 1 001**

**LEMBAR PENGESAHAN**

**APLIKASI PENTERJEMAH BAHASA INGGRIS KE BAHASA INDONESIA  
MENGUNAKAN NATURAL LANGUAGE PROCESSING (NLP)**

**SKRIPSI  
JURUSAN TEKNIK ELEKTRO**

Diajukan Untuk Memenuhi Persyaratan  
Memperoleh Gelar Sarjana Teknik

Disusun oleh:

**AGUNG NUGROHO PRAMUDHITA  
NIM. 0710630064-63**

Skripsi ini telah diuji dan dinyatakan lulus pada  
Tanggal 27 Januari 2012

**DOSEN PENGUJI**

**Waru Djuriatno, ST., MT.**  
**NIP.19690725 199702 1 001**

**R. Arief Setyawan, ST., MT.**  
**NIP.19750819 199903 1 001**

**Ali Mustafa, ST., MT.**  
**NIP. 19710601 200003 1 001**

Mengetahui,  
Ketua Jurusan Teknik Elektro

**Dr. Ir. Sholeh Hadi Pramono, MS**  
**NIP. 19580728 198701 1 001**



## KATA PENGANTAR

Puji syukur kepada Allah Tuhan YME yang telah memberikan segala nikmat dan rahmat-Nya sehingga penyusunan skripsi ini dapat diselesaikan. Karena hanya dengan pertolongan-Nya semata penulis mampu melewati segala kendala yang ada selama penyusunan skripsi ini.

Skripsi berjudul “*aplikasi penerjemah Bahasa Inggris ke Bahasa Indonesia menggunakan natural language processing (NLP)*” ini disusun sebagai salah satu syarat untuk mendapatkan gelar Sarjana Teknik di Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya. terselesaikannya skripsi ini tentunya tidak lepas juga dari bantuan berbagai pihak. Oleh sebab itu, dengan segala kerendahan hati penulis menyampaikan terima kasih kepada:

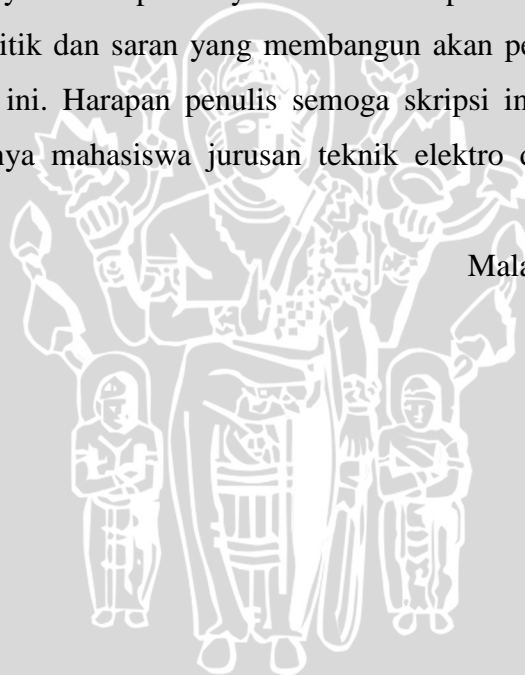
1. Kedua Orang Tua (Ayahanda Yoto dan Ibunda Sufairoh), serta seluruh keluarga penulis yang senantiasa memberikan do'a , nasihat dan motivasi yang tulus dalam menyelesaikan skripsi ini.
2. Bapak Dr. Ir. Sholeh Hadi Pramono, MS selaku Ketua Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya.
3. Bapak M. Azis Muslim, ST.,MT.,Ph.D selaku Sekretaris Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya.
4. Bapak Mochammad Rif'an , ST ,MT selaku Plt. Ketua Program Studi Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya.
5. Bapak Waru Djuriatno, ST., MT selaku KKDK Teknik Informatika dan Komputer Teknik Elektro yang telah memberikan arahan dalam pengambilan judul skripsi ini.
6. Bapak Ir. Muhammad Aswin, MT selaku dosen pembimbing I atas bantuan dan motivasi serta bimbingannya selama ini.
7. Bapak Adharul Muttaqin ST.,MT selaku dosen pembimbing II atas bantuan dan motivasi serta bimbingannya selama ini.
8. Bapak dan Ibu Dosen serta karyawan jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya atas kesediannya membagi ilmu.

9. Bapak Goegoes Dwi N ST., MT selaku dosen pembimbing akademik atas nasehatnya selama masa perkuliahan
10. Nourma Devita , Muhammad Saiful, Judis Priastono, Finsa Ferdifiansyah, serta Rizky Ramadhan atas bantuan , nasehat , dukungan dan persahabatannya.
11. Pak Fakhrudin, Mas Andik, mas Teguh dan mas Kuncoro atas bantuan dan dukungannya.
12. Teman-teman CORE 2007 , Teman-teman Asisten Laboratorium Sistem Informasi dan Komputer TEUB.dan Kiass yang setia menemani.
13. Semua pihak yang telah banyak membantu penyelesaian skripsi ini.

Penulis menyadari sepenuhnya bahwa skripsi ini masih banyak kekurangan. Segala kritik dan saran yang membangun akan penulis terima demi kesempurnaan skripsi ini. Harapan penulis semoga skripsi ini bermanfaat bagi pembaca dan khususnya mahasiswa jurusan teknik elektro dimasa yang akan datang.

Malang, Januari 2012

Penyusun



## DAFTAR ISI

KATA PENGANTAR .....	i
DAFTAR ISI.....	iii
DAFTAR GAMBAR .....	v
DAFTAR TABEL.....	vi
ABSTRAK .....	vii
BAB I.....	1
BAB II.....	6
2.1 Natural language processing (NLP) .....	6
2.1.1 Proses Dalam NLP.....	6
2.1.2 Komponen Utama Bahasa Alami .....	7
2.2 Part of Speech.....	8
2.2.1 <i>Verb</i> .....	8
2.2.2 <i>Noun</i> .....	9
2.2.3 <i>Adjective</i> .....	10
2.2.4 <i>Adverb</i> .....	10
2.2.5 <i>Pronoun</i> .....	11
2.2.6 <i>Preposition</i> .....	12
2.2.7 <i>Conjunction</i> .....	12
2.2.8 <i>Interjection</i> .....	12
2.2.9 <i>Article</i> .....	12
2.3 Main Tenses .....	13
2.3.1 <i>Present Tense</i> .....	13
2.3.2 <i>Past Tense</i> .....	15
2.3.3 <i>Future Tense</i> .....	17
2.4 Idiom .....	18
2.5 <i>Database</i> .....	19
2.6 DBMS – <i>Database Management System</i> .....	20
2.6.1 Pengertian DBM .....	20
2.6.2 Fungsi DBMS .....	21
2.6.3 Contoh DBMS .....	21
2.6.4 Keuntungan Penggunaan DBMS .....	22
2.6.5 Kerugian Penggunaan DBMS .....	22
2.7 SQL – <i>Structured Query Language</i> .....	23
2.7.1 Data Definition Language .....	23
2.7.2 Data Manipulation Language.....	23
2.8 Delphi .....	24
2.8.1 Area Kerja Borland Delphi 7 .....	25
2.8.2 Form .....	28
BAB III .....	31



3.1	Studi Literatur.....	31
3.2	Analisis Kebutuhan .....	31
3.3	Perancangan.....	31
3.4	Implementasi .....	32
3.5	Pengujian Sistem .....	32
3.6	Kesimpulan dan Saran.....	32
BAB IV .....		33
4.1.	Kebutuhan Sistem.....	33
4.1.1	Perangkat Keras .....	33
4.1.2	Perangkat Lunak .....	33
4.2.	Perancangan Aplikasi Secara Umum .....	34
4.3.	Pembuatan Aplikasi Penerjemah.....	34
4.3.1	<i>Database</i> Sistem .....	34
4.3.2	GUI Sistem .....	38
4.3.3	Cara Kerja Aplikasi Penerjemah .....	39
4.3.4	Proses Translasi .....	44
BAB V.....		54
5.1	Metode Pengujian.....	54
5.2	Hasil Pengujian.....	54
5.2.1.	<i>Past simple</i> .....	55
5.2.2.	<i>Past progressive</i> .....	55
5.2.3.	<i>Past perfect</i> .....	56
5.2.4.	<i>Past perfect progressiv</i> .....	57
5.2.5.	<i>Present simple</i> .....	58
5.2.6.	<i>Present progressive</i> .....	58
5.2.7.	<i>Present perfect</i> .....	59
5.2.8.	<i>Present perfect progressive</i> .....	59
5.2.9.	<i>Future simple</i> .....	60
5.2.10.	<i>Future progressive</i> .....	61
5.2.11.	<i>Future perfect</i> .....	61
5.2.12.	<i>Future perfect progressive</i> .....	62
5.3	Analisis hasil pengujian.....	62
BAB VI .....		64
5.1	Kesimpulan.....	64
5.2	Saran .....	64
DAFTAR PUSTAKA .....		66

## DAFTAR GAMBAR

Gambar 2.1 Area kerja delphi .....	26
Gambar 2.2 Menu utama.....	26
Gambar 2.3 <i>Toolbar</i> .....	27
Gambar 2.4 Form designer.....	27
Gambar 2.5 Object inspector.....	28
Gambar 2.6 Component palette .....	28
Gambar 4.1 Diagram Sistem .....	34
Gambar 4.2 Tabel Kamus .....	35
Gambar 4.3 Tabel Kalimat <i>Temporary</i> .....	36
Gambar 4.4 Tabel <i>Disjunct</i> .....	37
Gambar 4.5 Tabel <i>Disjunct</i> .....	37
Gambar 4.6 Tabel arti .....	38
Gambar 4.6 Tampilan GUI Aplikasi Penerjemah.....	38
Gambar 4.7 <i>flowchart</i> Cara Kerja Aplikasi Penerjemah.....	40
Gambar 4.8 <i>flowchart</i> metode pengosongan table <i>temporary</i> .....	41
Gambar 4.9 <i>flowchart</i> metode pengganti spasi .....	42
Gambar 4.10 <i>flowchart</i> metode penghitung jumlah kata .....	43
Gambar 4.11 <i>flowchart</i> proses penelusuran kalimat <i>input</i> .....	45
Gambar 4.12 <i>flowchart</i> fungsi cari frasa.....	46
Gambar 4.13 <i>flowchart</i> fungsi cari kata .....	47
Gambar 4.14 <i>flowchart</i> fungsi cari disjunct.....	48
Gambar 4.15 <i>flowchart</i> fungsi simpan tanda .....	49
Gambar 4.16 <i>flowchart</i> fungsi Imbuhan .....	51
Gambar 4.18 <i>flowchart</i> fungsi translate .....	53



## DAFTAR TABEL

Tabel 2.1 Contoh Penggunaan Verb .....	9
Tabel 2.2 Contoh penggunaan <i>Adjective</i> .....	10
Tabel 2.3. Contoh Penggunaan <i>Adverb</i> .....	11
Tabel 4.1 Keterangan <i>field</i> tabel kamus .....	35
Tabel 4.2 Keterangan <i>field</i> tabel kalimat <i>temporary</i> .....	36
Tabel 4.3 Keterangan <i>field</i> tabel <i>Disjunct</i> .....	37
Tabel 4.4 Keterangan <i>field</i> tabel Frasa.....	37
Tabel 4.5 Keterangan <i>field</i> tabel arti .....	38
Tabel 5.2 Hasil pengujian kalimat <i>past progressive</i> .....	55
Tabel 5.3 Hasil pengujian kalimat <i>past perfect</i> .....	56
Tabel 5.4 Hasil pengujian kalimat <i>past perfect progressive</i> .....	57
Tabel 5.6 Hasil pengujian kalimat <i>present progressive</i> .....	58
Tabel 5.7 Hasil pengujian kalimat <i>present perfect</i> .....	59
Tabel 5.8 Hasil pengujian kalimat <i>present perfect progressive</i> .....	59
Tabel 5.9 Hasil pengujian kalimat <i>future simple</i> .....	60
Tabel 5.10 Hasil pengujian kalimat <i>future progressive</i> .....	60
Tabel 5.11 Hasil pengujian kalimat <i>future perfect</i> .....	61
Tabel 5.12 Hasil pengujian kalimat <i>future Perfect Progressive</i> .....	62

## ABSTRAK

Bahasa Inggris saat ini merupakan bahasa yang paling banyak digunakan di berbagai negara dunia baik sebagai bahasa utama maupun bahasa sekunder. Oleh karena itu banyak sumber ilmu pengetahuan dan informasi baik itu di media cetak maupun di media elektronik yang menggunakan Bahasa Inggris sebagai bahasa pengantarnya. Hal ini terjadi seiring dengan meningkatnya kebutuhan akan informasi dan berkembangnya ilmu pengetahuan. Fenomena tersebut menimbulkan kesulitan bagi sebagian besar masyarakat Indonesia terutama pelajar dan mahasiswa yang kemampuan Bahasa Inggrisnya masih rendah. Karena itu dibutuhkan sebuah cara untuk dapat membantu menerjemahkan sumber informasi dan ilmu pengetahuan tersebut.

Pada skripsi ini dijelaskan pembuatan sebuah aplikasi untuk menerjemahkan Bahasa Inggris ke Bahasa Indonesia yang diharapkan dapat membantu dalam menerjemahkan berbagai literatur ilmu pengetahuan dan sumber informasi. Aplikasi ini dibuat menggunakan *delphi 7* dan *MySQL*. Dalam proses pembuatannya digunakan metode *Morphological Analysis* dan *Syntactic Analysis* yang terdapat dalam *Natural Language Processing (NLP)* dimana metode ini memungkinkan untuk dapat diperoleh hasil terjemahan yang lebih akurat karena diperhatikan tiap kata yang ada, juga dianalisis jenis katanya, dan diberi perlakuan sesuai tata bahasa yang ada.

Pada proses pengujian diketahui bahwa proses penerjemahan dapat berjalan dengan baik pada tiap jenis kalimat utama (*main tense*) yang terdapat dalam Bahasa Inggris. Pada proses penerjemahan juga terdapat metode untuk mendeteksi kesalahan *syntax pada* kalimat *input*, ketika terjadi kesalahan *syntax* maka akan terdapat *caption* yang menyatakan bahwa *user* memberikan kalimat Bahasa Inggris yang salah. Dalam program yang dibuat juga terdapat terjemahan alternatif apabila terdapat makna ganda pada suatu kalimat, *user* akan dapat memilih terjemahan yang dianggap benar sehingga jika terdapat kalimat yang sama muncul kembali maka hasil terjemahan yang muncul adalah terjemahan yang benar.

**Kata Kunci:** Bahasa Inggris, Bahasa Indonesia, *delphi 7*, *MySQL*, NLP



## BAB I

### PENDAHULUAN

#### 1.1 LATAR BELAKANG

Manusia adalah makhluk sosial, hal ini berarti manusia saling membutuhkan antara satu manusia dengan manusia yang lainnya. Atas dasar saling membutuhkan itulah terjadi interaksi antar manusia. Dalam interaksi tersebut manusia saling berkomunikasi untuk menyampaikan apa yang diinginkan ataupun dipikirkan manusia tersebut. Untuk dapat berkomunikasi dengan baik diperlukan alat yang bernama bahasa. Sejak zaman purba manusia telah mengenal bahasa berupa bahasa tubuh, gambar atau simbol-simbol, kemudian seiring dengan berkembangnya peradaban manusia, manusia mulai menemukan cara untuk dapat berkomunikasi dengan cara yang lebih efektif dan efisien dengan menggunakan bahasa lisan maupun bahasa tulis. Bahasa sendiri menurut Walija (1996:4) ialah komunikasi yang paling lengkap dan efektif untuk menyampaikan ide, pesan, maksud, perasaan dan pendapat kepada orang lain. Pendapat lainnya tentang definisi bahasa diungkapkan oleh Syamsuddin (1986:2), beliau memberi dua pengertian bahasa. Pertama, bahasa adalah alat yang dipakai untuk membentuk pikiran dan perasaan, keinginan dan perbuatan-perbuatan, alat yang dipakai untuk mempengaruhi dan dipengaruhi. Kedua, bahasa adalah tanda yang jelas dari kepribadian yang baik maupun yang buruk, tanda yang jelas dari keluarga dan bangsa, tanda yang jelas dari budi kemanusiaan.

Terdapat banyak ragam bahasa yang digunakan oleh manusia, bahasa yang digunakan tersebut pada umumnya tergantung dari lokasi geografis atau negara tempat manusia tersebut tinggal maupun ras atau suku dari manusia tersebut. Karena ragam dari bahasa tersebut cukup banyak, *pasti* terjadi kendala ketika seseorang dari negara tertentu berkomunikasi dengan orang dari negara lain yang menggunakan bahasa yang berbeda. Oleh karena hal tersebut maka diperlukan bahasa



internasional yang dapat digunakan untuk berkomunikasi antara manusia yang satu dengan yang lainnya walaupun berbeda lokasi geografis atau negara maupun berbeda suku atau ras. Saat ini bahasa internasional yang digunakan di dunia adalah Bahasa Inggris.

Dengan digunakannya Bahasa Inggris sebagai bahasa internasional maka sumber informasi dan ilmu pengetahuan baik buku-buku, artikel-artikel, dan berbagai literatur terkini yang tersedia di dunia maupun di internet kebanyakan menggunakan Bahasa Inggris. Padahal kebutuhan akan informasi dan ilmu pengetahuan terus menerus meningkat. Oleh karena itu Bahasa Inggris mutlak diperlukan. Hal ini menimbulkan kendala bagi kebanyakan orang di Indonesia yang pada kenyataannya sebagian besar tidak mampu berbahasa Inggris. Kendala tersebut berupa sulitnya menerjemahkan sumber informasi atau ilmu pengetahuan yang berbahasa Inggris ke bahasa yang dipahami, yaitu Bahasa Indonesia.

Berdasarkan masalah di atas maka dibutuhkan *software* yang dapat membantu menerjemahkan Bahasa Inggris ke Bahasa Indonesia. Untuk mengakomodir hal tersebut telah ada *software* berbasis web yang disediakan secara gratis oleh *google* yang mampu menerjemahkan dari Bahasa Inggris ke Bahasa Indonesia walaupun masih terdapat beberapa kekurangan. Salah satu kekurangan dari *software* ini adalah hasil terjemahan yang kurang baik dan kurang akurat dalam hal kesesuaian dengan makna sebenarnya. *Software* ini juga membutuhkan koneksi internet bagi siapapun yang ingin menggunakannya hal ini tentu menjadi masalah bagi sebagian besar orang, karena ketersediaan internet di Indonesia belum mencakup semua orang dan biaya internet masih tergolong mahal, jadi tidak semua orang dapat menggunakan *software* ini. Selain *software* tersebut di atas ada juga *software* yang bisa diinstall kedalam sistem komputer sehingga tidak memerlukan koneksi internet, tetapi harga dari *software* itu sendiri sangat mahal dan tidak terjangkau bagi sebagian besar orang selain itu juga *software* tersebut dapat merusak konfigurasi sistem sehingga mengakibatkan komputer sering hang.

*Software* tersebut juga kurang memperhatikan struktur dan tata bahasa yang berlaku, sehingga hasil terjemahan kurang baik.

Berdasarkan uraian diatas maka penyusun membuat suatu *software* penerjemah Bahasa Inggris ke Bahasa Indonesia yang nantinya akan bisa dijangkau oleh masyarakat luas. *Software* penerjemah ini juga dibuat sebisa mungkin dapat mengakomodir idiom serta lebih memperhatikan struktur bahasa dan tata bahasa yang ada sehingga hasil terjemahan dapat menjadi lebih baik lagi

## 1.2 RUMUSAN MASALAH

Adapun rumusan masalah yang terdapat dalam pembuatan *software* ini adalah sebagai berikut:

1. Bagaimana struktur *database* yang digunakan program penerjemah?
2. Bagaimana program penerjemah bisa mengenali idiom atau frasa?
3. Bagaimana program penerjemah dapat mengaplikasikan penggunaan tata bahasa yang benar ?

## 1.3 BATASAN MASALAH

Mengacu pada permasalahan yang telah dirumuskan, maka hal-hal yang berkaitan dengan *software* yang akan dibuat, diberi batasan sebagai berikut:

1. Sistem hanya dapat menerjemahkan dari Bahasa Inggris ke Bahasa Indonesia.
2. Sistem hanya dapat menerima *input* berupa satu kalimat.
3. Sistem hanya dapat menerjemahkan 12 tenses utama.m
4. Sistem hanya dapat menerjemahkan secara tekstual



#### 1.4 TUJUAN

Tujuan dari penulisan tugas akhir(skripsi) ini adalah:

Merancang dan membuat aplikasi penerjemah Bahasa Inggris ke Bahasa Indonesia yang memperhatikan struktur dan tata bahasa dalam proses menerjemahkannya dan juga bisa mengakomodir idiom

#### 1.5 MANFAAT

Manfaat yang diharapkan dapat diperoleh melalui pengerjaan skripsi ini adalah :

##### 1.5.1. Bagi Penyusun :

- a. Penyusun dapat merancang dan membuat aplikasi untuk menerjemahkan Bahasa Inggris ke Bahasa Indonesia
- b. Menerapkan ilmu yang telah diperoleh dari Teknik Elektro Konsentrasi Teknik Informatika dan Komputer Universitas Brawijaya.
- c. Menambah Kemampuan dan pemahaman tentang DBMS dan Pemrograman Delphi

##### 1.5.2. Bagi Pengguna :

- a. Dapat memanfaatkan aplikasi yang dibuat untuk membantu dalam pekerjaan ataupun mempelajari literatur yang menggunakan Bahasa Inggris.

##### 1.5.3. Bagi Jurusan:

- a. Sebagai bahan referensi dan sebagai bahan kajian lebih lanjut bagi peneliti berikutnya.

#### 1.6 SISTEMATIKA PENULISAN

Sistematika penulisan dan gambaran untuk setiap bab pada laporan skripsi ini adalah sebagai berikut :



**BAB I           Pendahuluan**

Menjelaskan latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan skripsi.

**BAB II           Dasar Teori**

Menjelaskan kajian pustaka dan dasar teori yang digunakan.

**BAB III          Metodologi**

Menjelaskan metode yang digunakan dalam pengerjaan skripsi.

**BAB IV          Perancangan**

Menjelaskan analisis kebutuhan dan perancangan perangkat lunak, menerapkan teknologi yang akan digunakan.

**BAB V           Perancangan dan Implementasi**

Menjelaskan langkah-langkah pembuatan Aplikasi Penerjemah Bahasa Inggris Ke Bahasa Indonesia

**BAB VI          Kesimpulan dan Saran**

Berisi Kesimpulan dan *Saran*



## BAB II

### DASAR TEORI

Pada bab ini dibahas mengenai dasar teori yang digunakan untuk menunjang penulisan skripsi mengenai Aplikasi Penerjemah Bahasa Inggris Ke Bahasa Indonesia. Beberapa dasar teori yang dimaksud diantaranya adalah NLP, *Part of Speech*, *Tenses*, *Idiom*, *Database*, *DBMS*, *SQL*, *Borland Delphi 7*

#### 2.1 Natural language processing (NLP)

Pengolahan bahasa alami adalah upaya untuk mengekstrak makna *representasi* dari teks bebas. Hal ini dilakukan untuk mencari tahu siapa yang melakukan apa kepada siapa, kapan, di mana, bagaimana dan mengapa. NLP biasanya membuat penggunaan konsep-konsep linguistik seperti *part of speech* (*noun*, *Verb*, *Adjective*, dll) dan struktur tata bahasa baik direpresentasikan sebagai ungkapan-ungkapan seperti nomina frase atau frase preposisi, atau hubungan ketergantungan seperti subjek atau objek (Anne Kao, 2007). Pada prinsipnya bahasa alami adalah suatu bentuk *representasi* dari suatu pesan yang ingin dikomunikasikan antar manusia. Bentuk utama *representasinya* adalah berupa suara/ucapan (*spoken language*), tetapi sering pula dinyatakan dalam bentuk tulisan.

Inti dari pemrosesan bahasa alami adalah penguraian kalimat atau sering disebut dengan *parser*. *Parser* berfungsi untuk membaca kalimat, kata demi kata dan menentukan jenis kata apa saja yang boleh mengikuti kata tersebut.

##### 2.1.1. Proses Dalam NLP

pemrosesan bahasa alami dapat dibagi atas beberapa proses:

- a. *Morphological analysis*, dimana kata secara individu dianalisis ke dalam komponen-komponennya dan yang bukan kata (seperti tanda baca) dipisahkan dari kata
- b. *Syntactic analysis*, urutan linier dari kata ditransformasikan ke dalam struktur yang menunjukkan bagaimana kata saling berhubungan. Urutan kata yang melanggar aturan bahasa dalam pengkombinasian kata, akan ditolak
- c. *Semantic analysis*, dibuat mapping antara struktur sintaksis & objek dalam domain kerja. Struktur dimana tidak ada mapping yang memungkinkan, akan ditolak.
- d. *Discourse integration*, dimana arti dari kalimat secara individu bergantung pada kalimat-kalimat yang mendahului & mungkin dapat mempengaruhi arti dari kalimat-kalimat yang mengikutinya
- e. *Pragmatic analysis*, dimana struktur yang merepresentasikan apa yang diucapkan, diinterpretasikan kembali untuk menentukan apa arti sebenarnya.

### 2.1.2. Komponen Utama Bahasa Alami

Pengolahan bahasa alami terdiri dari dua bagian utama, yaitu : parser, sistem representasi pengetahuan dan pengolahan *output*.

- a. Parser  
Suatu sistem yang mengambil kalimat *input* bahasa alami dan menguraikannya ke dalam beberapa bagian gramatikal (kata benda, kata kerja, kata sifat, dan lain-lain).
- b. Sistem Representasi Pengetahuan  
Suatu sistem yang menganalisis *output* parser untuk menentukan maknanya.



c. *Pengolahan Output*

Suatu terjemahan yang merepresentasikan sistem pengetahuan dan melakukan langkah- langkah yang bisa berupa jawaban atas bahasa alami atau *output* khusus yang sesuai dengan program komputer lainnya.

## 2.2 *Part of Speech*

*Parts of Speech* dalam Bahasa Inggris berarti jenis-jenis kata atau kelas-kelas kata. Disebut *Parts of Speech* – bagian-bagian dari ucapan atau bagian-bagian kalimat (pada dasarnya kalimat dalam bahasa adalah kalimat yang diucapkan, bukan dituliskan) – karena kata-kata ini merupakan suatu sistem yang diperlukan untuk membentuk sebuah kalimat, tanpa melihat apa tugas atau fungsinya masing-masing. Jadi, kata-kata itu merupakan “bahan dasar” di dalam sebuah “bangunan” kalimat, bukan “fungsi” kata di dalam kalimat. Mengapa demikian? Karena sebuah kata bisa mempunyai beberapa (lebih dari satu) fungsi dalam kalimat. Secara lengkap, ada 9 jenis kata atau kelas kata dalam Bahasa Inggris, yaitu:

### 2.2.1. **Verb**

*Verb*, verba atau kata kerja disebut “King” (raja), karena merupakan bagian terpenting dalam Bahasa Inggris. Setiap kalimat harus mengandung paling sedikit satu verb. Bisa dikatakan, tidak ada kalimat tanpa verb. Jenis-jenis kata lain boleh tidak ada, tetapi verb harus ada dalam setiap kalimat. Bahkan satu verb saja sudah cukup menjadi sebuah kalimat, yaitu dalam imperative sentence (kalimat perintah). Misalnya: ‘Stop!’ (Berhenti!), ‘Go!’ (Pergi!), ‘Listen!’ (Dengarkan!) dsb.

Dalam Bahasa Indonesia, verb sering disamakan dengan predikat. Pemahaman ini tidak sepenuhnya salah, karena dalam pola-pola kalimat Bahasa Indonesia harus selalu ada predikat, sedangkan

dalam verb patterns (pola-pola kalimat Bahasa Inggris) harus selalu ada verb. Misalnya:

Tabel 2.1 Contoh Penggunaan Verb

No	Kalimat	Pola
1	This is Fay Ahmed (Ini Fay Ahmed)	(S-V-C)/ (S-P)
2	Budi is a teacher (Budi (adalah seorang)( guru)	(S-V-C)/ (S-P)
3	Toni goes to school everyday (Toni pergi ke sekolah setiap hari).	(S-V-Adv/place-Adv/time)/.”(S-P-Ket/tempat-Adv/waktu).

**KETERANGAN:**

Bahasa Inggris:

S = Subject, Verb = *Verb*, C = Complement, Adv/ place = *Adverb* of place, Adv/time = *Adverb* of time, clause = klausa = anak kalimat.

Bahasa Indonesia:

S = Subyek (pelaku), P = Predikat = Ket/ tempat (*Averbia* tempat), Ket/waktu = Keterangan waktu (*Adverbia* waktu), anak klmt = anak kalimat (klausa).

**2.2.2. Noun**

*Noun* atau kata benda ialah sebuah kata yang digunakan untuk menamai orang, tempat, benda atau sebuah gagasan.

Nama Orang : George, Mary, Bill, Tom Scott.

Nama tempat : New York City, Midwest, Mexico.

Nama benda : book, chair, sun, cloud, bicycle

Gagasan : justice, courage, hope, love.

### 2.2.3. Adjective

*Adjective* atau kata sifat ialah sebuah kata yang digunakan untuk membatasi (“Modify”) atau menjelaskan kata benda (*Noun*) atau sebuah kata ganti (*Pronoun*). Kata “Modify” berarti “merubah”. Untuk merubah sebuah kata maksudnya merubah makna kata dengan membuat maknanya lebih jelas. Untuk mengetahui apakah sebuah kata merupakan kata sifat yang berfungsi menjelaskan atau merubah makna sebuah kata benda atau kata ganti bisa diketahui dengan menjawab pertanyaan-pertanyaan berikut ini:

- Jenisnya apa?
- Yang mana?
- Ada berapa jumlahnya?

Perhatikan bagaimana kata sifat yang dicetak miring dibawah ini menjawab pertanyaan-pertanyaan mengenai kata benda dan kata ganti yang dijelaskannya.

Tabel 2.2 Contoh penggunaan *Adjective*

No.	Jenisnya Apa?	Yang mana?	Ada berapa jumlahnya?
1	<i>A blue dress</i>	<i>That flower</i>	<i>Four times</i>
2	<i>A false note</i>	<i>Third base</i>	<i>Several girls</i>
3	<i>A rich lawyer</i>	<i>The broken bone</i>	<i>Each player</i>

### 2.2.4. Adverb

Sebuah *adverb* merupakan sebuah kata yang digunakan untuk menjelaskan kata kerja, sebuah kata sifat atau kata keterangan yang lain. Contohnya ada pada kata yang dicetak miring dibawah ini. Tanda yang paling mudah untuk mengenali suatu kata adalah *adverb* atau kata keterangan (*adverb of manner*) biasanya dibentuk dengan menambahkan kata akhiran -ly pada



kata sifat. Contohnya kata *great* yang dirubah menjadi kata *greatly*, dan kata *beautiful* dirubah menjadi *beautifully*. (Catatan: perhatikan bahwa beberapa kata yang berakhiran *-ly*, seperti kata *friendly* dan *lovely*, bukanlah kata keterangan, tetapi kata sifat, yang mana dalam kasus ini akar katanya berasal dari kata benda. Ada juga kata sifat yang tidak bisa dilacak asal usulnya yang ber-suffiks *-ly*, seperti kata *holy* dan *silly*. Cara untuk mengetahui bahwa satu kata masuk dalam kategori *adverb* bisa diketahui dengan memberikan beberapa pertanyaan seperti : *Where, When, How* dan *To what extent*. contoh *adverb* yang dapat dilihat tabel berikut ini:

Tabel 2.3. Contoh Penggunaan *Adverb*

No	WHERE	WHEN	HOW	TO WHAT EXTENT
1	<i>Play here</i>	<i>Play now</i>	<i>Play well</i>	<i>Do not play</i>
2	<i>He drove there</i>	<i>He drives daily</i>	<i>He drives carefully</i>	<i>He drove far</i>
3	<i>I fell down</i>	<i>I often fall</i>	<i>3 I fell clumsily</i>	<i>I almost fell</i>

### 2.2.5. Pronoun

Dalam ilmu linguistik dan tata bahasa, yang disebut “*pronoun*” (dalam bahasa Latin: *pronomen*) adalah sebuah kata yang digunakan untuk menjadi substitusi atau menggantikan kata benda atau frasa nomina dengan atau tanpa determiner, seperti kata “*you*” dan “*they*” dalam Bahasa Inggris. Frasa yang digantikan disebut sebagai anteseden dari *pronoun*. Sebagai contohnya mari kita kaji kalimat “*Lisa gave the coat to Phil.*” Ketiga kata benda yang ada di kalimat tersebut dapat digantikan oleh *pronouns* :”*She gave it to him.*” Jika kata “*the coat, Lisa dan Phill*” telah disebutkan lebih awal, pendengar dapat menyimpulkan bahwa pronomina “*she, it dan him*” mengacu

kepada ketiga kata benda tersebut sehingga makna kalimat bisa dipahami.

#### 2.2.6. Preposition

Sebuah *preposition* atau kata depan merupakan suatu kata yang digunakan untuk menunjukkan hubungan antara kata benda atau sebuah kata ganti. Hubungan diantara keduanya dapat dilihat dalam rangkai kalimat di bawah ini:

*I walked to the house* (Saya telah berjalan ke rumah)

*I walked around the house* (Saya telah berjalan mengelilingi rumah)

*I walked through the house* (Saya telah berjalan melewati rumah)

#### 2.2.7. Conjunction

Konjungsi ialah kata yang merupakan penghubung secara sintaksis terhadap konstituen yang lebih besar dan juga menyatukan hubungan semantik diantara keduanya. Sebuah konjungsi merupakan penghubung yang posisinya telah tetap atas satu atau dua unsur yang berkaitan, yang membedakannya dengan konstituen seperti kata keterangan konjungtif dalam Bahasa Inggris.

#### 2.2.8. Interjection

Sebuah interjeksi merupakan kata atau frasa yang menunjukkan emosi atau keterkejutan yang tidak mempunyai hubungan gramatikal terhadap kata manapun yang menjadi bagian kalimat. Mereka sering berada dalam punctuation yang menunjukan eksklamasi yang jarang digunakan. Lebih baik menghindari penggunaan ini dalam penulisan formal kecuali dalam kutipan langsung. Dalam penggunaan kebanyakan kamus penyingkatan dengan “interj.” berarti “interjection.” Atau ‘interjeksi. Misalnya kata : Ouch, Hey, Oh my, Wow, psst, Ugh, excellent, well, well.

### 2.2.9. Article

Artikel (sering disingkat dengan ART) atau kata sandang merupakan kata yang dikombinasikan dengan kata benda untuk menunjukkan tipe refensi yang dibuat oleh kata benda. Artikel menspesifikasikan kepastian gramatikal kata benda yang dalam bahasa tertentu meluas menjadi cakupan volume dan jumlah angka. Artikel dalam Bahasa Inggris ialah kata *the*, *a*, dan *an*. (kata *Some* dapat dalam keadaan tertentu berfungsi sebagai bentuk jamak dari “a atau “an”).

### 2.3 Main Tenses

Ada tiga jenis kalimat atau tenses utama (main tenses) yang perlu diketahui.

- *Present tense*
- *Past tense*
- *Future tense*

Main *tense* di atas masing-masing bisa dibagi menjadi empat bagian, yaitu bentuk sederhana (*simple*), bentuk sedang (*Continuous*), dan bentuk sempurna (*perfect*).

#### 2.3.1. Present Tense

##### a. *Simple Present tense*

Digunakan untuk menunjukkan suatu kebiasaan (peristiwa yang terjadi berulang-ulang). Contohnya: Dia belajar Fisika dan Matematika setiap hari Kamis.

*Simple Present tense* memiliki struktur :

*Subject + verb (dalam bentuk infinitive) + ...*

Contoh: She studies Physics and Mathematics every Thursday.



Sementara, bentuk kalimat tanya dirumuskan sebagai berikut.

*Do/does + subject + verb + ...*

Contoh: Does She study Physics and Mathematics every Thursday?

**b. Present Continous Tense**

Digunakan untuk menunjukkan suatu peristiwa yang sedang terjadi. Contohnya: Ibu saya sedang menyiram bunga sekarang.

*Present Continous tense* memiliki struktur:

*Subject + am/is/are + verb – ing (Present participle) + ...*

Contoh: Mother is watering flowers now.

Sementara, struktur kalimat tanya *tense* ini dirumuskan:

*Am/is/are + subject + verb – ing + ...*

Contoh: Is mother watering flowers now?

**c. Present Perfect Tense**

Digunakan untuk menunjukkan suatu peristiwa yang terjadi pada masa lalu, tetapi masih berlangsung sampai sekarang. Contoh: Keluarga kami sudah menempati rumah ini selama 10 tahun.

*Present perfect tense* memiliki struktur:

*Subject + have / has + past participle + ....*

Contoh : our family have occupied this house for 10 years.

**d. Present Perfect Continous Tense**

Digunakan untuk menunjukkan suatu peristiwa masa lalu yang masih berlangsung sekarang dan akan diteruskan pada waktu

yang akan datang. Contoh: Kakak saya sudah bekerja di Jepang selama 7 tahun.

*Present Continous* tense, dirumuskan sebagai berikut:

*Subject + have / has + been + verb – ing + ...*

Contoh: My brother been working in Japan for 7 years.

### 2.3.2. Past Tense

#### a. *Simple Past Tense*

Digunakan untuk menunjukkan peristiwa yang terjadi di masa lalu. Contoh: Veronica membeli rumah ini 3 tahun yang lalu.

*Simple past tense* memiliki struktur:

*Subject + verb (bentuk lampau) + ....*

Contoh: Veronica bought this house 7 years ago.

Sementara bentuk kalimat tanya *tense* ini, dirumuskan :

*Did + subject + verb (dalam bentuk pertama) + ....*

Contoh : Did Veronica buy this house 7 years ago?

#### b. *Past Continous Tense*

Digunakan untuk menunjukkan suatu peristiwa yang sedang berlangsung pada waktu tertentu di masa lalu. Contoh: Saya sedang membersihkan rumah pada pukul 9.00 kemarin pagi.

*Past Continous tense* dirumuskan sebagai berikut :

*Subject + was/were + verb-ing (Present participle) + ...*

Contoh : I was cleaning the house at 9.00 o'clock yesterday morning.

**c. Past Perfect Tense**

Digunakan untuk menunjukkan suatu peristiwa yang terjadi pada masa lalu. Lalu, diikuti peristiwa lain sebagai keterangan atas kejadian pertama yang sudah selesai. Contoh: Dia sudah menjual mobil itu ketika saya datang untuk melihatnya kemarin sore.

*Past Perfect tense* memiliki struktur :

*Subject + had + past participle ...*

*Simple Past tense* memiliki struktur :

*Subject + verb (past form) + ...*

Lalu, kedua rumusan di atas disambung dengan kata penghubung seperti “when, before, atau after.”

Contoh : He had sold the car when I came to see it yesterday afternoon.

**d. Past Perfect Continous Tense**

Digunakan untuk menunjukkan sebuah kejadian yang terjadi di masa lalu, tetapi masih berlangsung hingga saat ini dan mendatang. Contoh: Pada pukul 9.00 tadi pagi, Ratih mulai mengetik laporan bulanan di kantornya. Dua jam kemudian, laporan bulanan itu belum selesai diketiknya dan dia masih harus mengetik sampai selesai.

Kalimat di atas akan lebih simpel jika ditulis dalam *past perfect tense* dengan struktur :



*Subject + had + been + verb-ing (Present participle) + ....*

Contoh : Diana had been typing the monthly report for 2 hours at 11 o'clock this morning.

### 2.3.3. Future Tense

#### a. *Simple Future Tense*

Digunakan untuk menunjukkan suatu peristiwa yang terjadi di masa yang akan datang. Contoh: Nyonya Lia akan memberiku hadiah istimewa besok.

Struktur *simple future tense* dirumuskan :

*Subject + will + verb (infinitive) + ...*

Contoh : Mrs. Lia will give me special gift tomorrow.

#### b. *Future Continous Tense*

Digunakan untuk menunjukkan suatu peristiwa yang telah direncanakan. Contoh: Aku akan menulis surat nanti malam.

Struktur *future Continous tense* dirumuskan :

*Subject + am/is/are + going to + verb (infinitive tanpa to) + ...*

Contoh: I am going to write a letter tonight.

#### c. *Future Continous Tense*

Digunakan untuk menunjukkan suatu kegiatan yang telah direncanakan pada waktu tertentu di masa yang akan datang. Contoh: Sarah akan meninggalkan rumah nenek saat kakeknya pulang jam 5 sore ini.

Struktur *perfect Continous tense* biasa dirumuskan :

*Subject + shall/will + be + verb-ing (Present participle) + ...*

Contoh: Sarah will be leaving her grandmother's house if her grandfather come home at 5.00 o'clock this afternoon.

#### d. **Future Perfect Tense**

Digunakan untuk menunjukkan suatu kegiatan yang sudah selesai dilakukan pada masa yang akan datang. Contoh: Fita sudah akan menyelesaikan tugasnya sebelum bulan Januari.

Struktur *future perfect tense* dirumuskan sebagai berikut:

*Subject + will + have + past participle + ...*

Contoh: Fita will have completed her work by the end of January.

#### e. **Future Perfect Continuous Tense**

Digunakan untuk menunjukkan sebuah kejadian yang dimulai pada waktu lalu, tetapi masih tetap berlangsung sampai waktu yang akan datang. Contoh: Ketika Lala datang pagi ini, aku *pasti* masih tidur selama 1 jam.

Struktur *future perfect Continuous tense*:

*Subject + shall/will + have + been + verb-ing + ...*

Contoh: When Lala arrives this morning, I will have been sleeping for 1 hour.

### 2.4 Idiom

Idiom adalah suatu ungkapan (seperti istilah atau frase) yang maknanya tak dapat diturunkan dari definisi langsung dan penyusunan bagian-bagiannya, melainkan merupakan suatu makna tak langsung yang

hanya dikenal melalui penggunaan umum. Dalam linguistik, idiom umumnya dianggap merupakan gaya bahasa yang bertentangan dengan prinsip penyusunan (*principle of compositionality*), walaupun masih terjadi perdebatan mengenai hal tersebut.

Contoh idiom:

Good morning = selamat pagi

to learn by heart = menghafal

to have a swim = to swim = berenang

to have break = istirahat

## 2.5 Database

Terdapat beberapa definisi tentang *Database* itu sendiri diantaranya yaitu: :

- **Database Menurut Gordon C. Everest:**

Menurut Gordon C. Everest (1986) menjelaskan bahwa *Database* adalah koleksi atau kumpulan data yang mekanis, terbagi/*shared*, terdefinisi secara formal dan dikontrol terpusat pada organisasi.

- **Database Menurut C.J. Date:**

*Database* adalah koleksi “data operasional” yang tersimpan dan dipakai oleh sistem aplikasi dari suatu organisasi (C.J. Date, 1985). Sedangkan data operasional tersebut meliputi:

- a. *Data input* : adalah data yang masuk dari luar sistem
- b. *Data output* : adalah data yang dihasilkan sistem
- c. *Data operasional* : adalah data yang tersimpan pada sistem

- **Database Menurut S. Attre:**

Menurut S. Attre dalam Korth (1991) *database* adalah koleksi data-data yang saling berhubungan mengenai suatu organisasi/*enterprise* dengan macam-macam pemakaiannya.



- **Database Menurut Richard Stones Neil Matthew:**

*Database* adalah suatu kumpulan informasi yang disimpan secara sistematis sehingga mempermudah dalam pengelolaan dan pencariannya (Richard Stones Neil Matthew, 2005)

Dari beberapa definisi dari para ahli diatas dapat disimpulkan bahwa yang dimaksud dengan *database* merupakan sistem penyimpanan koleksi data yang saling berhubungan secara sistematis sehingga mempermudah pengolahannya. Sistematis dari suatu *database* berasal dari cara pengorganisasian data sesuai dengan model *database*-nya. Model *database* yang sering digunakan saat ini adalah model relational *database*.

Suatu *database* komputer akan disimpan dalam media penyimpanan. Media penyimpanan tersebut dapat kita ibaratkan sebuah storage penyimpanan, misalnya hardisk. Dalam *database*, data yang ada tidak hanya diletakkan dan disimpan begitu saja dalam sebuah media penyimpanan, akan tetapi dikelola dengan sebuah sistem pengaturan *database* yang sering disebut dengan Database Management System (DBMS). Dengan begitu, suatu data dengan jumlah besar dan kompleks dapat tersusun sangat baik sehingga memungkinkan pengaksesan data dengan mudah dan cepat oleh pengguna

## 2.6 DBMS – Database Management System

### 2.6.1. Pengertian DBMS :

DBMS mempunyai beberapa pengertian, Menurut C.J. Date (1985) DBMS adalah merupakan *software* yang menhandel seluruh akses pada *database* untuk melayani kebutuhan *user*. Sedangkan menurut S, Attre dalam Korth (1991) DBMS adalah *software*, hardware, firmware dan procedure-procedure yang memanager *database*. Firmware adalah *software* yang telah menjadi modul yang tertanam pada hardware (ROM). Gordon C. Everest (1986) mengatakan bahwa: DBMS adalah manajemen yang efektif

untuk mengorganisasi sumber daya data. Jadi dapat disimpulkan DBMS adalah semua peralatan komputer (Hardware + *Software* + Firmware) yang digunakan untuk mengorganisasi data.

DBMS dilengkapi dengan bahasa yang berorientasi pada data (High level data language) yang sering disebut juga sebagai bahasa generasi ke 4 (fourth generation language).

#### 2.6.2. Fungsi DBMS :

DBMS merupakan *software* (dan hardware) yang khusus didesain untuk melindungi dan memanager *database*. Dengan menggunakan DBMS, maka dapat :

- a. Mendefinisikan data dan hubungannya.
- b. Mendokumentasikan struktur dan definisi data
- c. Menggambarkan, mengorganisasikan dan menyimpan data untuk akses yang selektif/dipilih dan efisien.
- d. Hubungan yang sesuai antara *user* dengan sumber daya data.
- e. Perlindungan terhadap sumber daya data akan terjamin, dapat diandalkan, konsisten dan benar.
- f. Memisahkan masalah Logical dan physical sehingga merubah implementasi *database* secara fisik tidak menghendaki *user* untuk merubah maksud data (Logical).
- g. Menentukan pembagian data kepada para *user* untuk mengakses secara concurrent pada sumber daya data.

#### 2.6.3. Contoh DBMS :

- a. *Database Hierarchy* : Pengaksesan data harus mengikuti aturan hierarchy yang sudah didefinisikan terlebih dahulu. Contoh : IMS-2 (Information Management System) oleh IBM, 1968



- b. Data Network : Data membentuk jaringan yang lebih bebas dari model hierarchy. Contoh : IDMS (Integrated Database Management System) oleh Cullinet Software Inc, 1972
- c. Data Relational : Data dikelompokkan secara bebas menurut jenisnya lewat proses normalisasi. Contoh : – INGRES oleh UN of CA & Relational Tech., 1973. System-R oleh IBM Research, 1975. ORACLE oleh Relational Software Inc. , 1979

#### 2.6.4. Keuntungan Penggunaan DBMS

Penggunaan DMBS untuk mengelola data mempunyai beberapa keuntungan, yaitu :

- a. Kebebasan data dan akses yang efisien
- b. Mereduksi waktu pengembangan aplikasi
- c. Integritas dan keamanan data
- d. Administrasi keseragaman data
- e. Akses bersamaan dan perbaikan dari terjadinya crashes (tabrakan dari proses serentak).
- f. Mengurangi *data redundancy* : Data redundansi dapat direduksi/dikurangi, tetapi tidak dapat dihilangkan sama sekali (untuk kepentingan *keyfield*)
- g. Membutuhkan sedikit memory untuk penyimpanan data

#### 2.6.5. Kerugian penggunaan DBMS

Penggunaan DMBS untuk mengelola data mempunyai beberapa kerugian ,antara lain :

- a. Memperoleh perangkat lunak yang mahal (teknologi DBMS, *Operation, Conversion, Planning, Risk*). DBMS *mainframe* masih sangat mahal. DBMS berbasis mikro biayanya mencapai beberapa ratus dolar, dapat menggambarkan suatu organisasi yang kecil secara berarti



- b. Memperoleh konfigurasi perangkat keras yang besar. DBMS sering memerlukan kapasitas penyimpanan primer dan sekunder yang lebih besar daripada yang diperlukan oleh program aplikasi lain. Juga, kemudahan yang dibuat oleh DBMS dalam mengambil informasi mendorong lebih banyak terminal pemakai yang disertakan dalam konfigurasi daripada jika sebaliknya.
- c. Mempekerjakan dan mempertahankan staf DBA DBMS memerlukan pengetahuan khusus agar dapat memanfaatkan kemampuan secara penuh. Pengetahuan khusus ini paling baik diberikan oleh pengelola *database*.

## 2.7 SQL – Structured Query Language

*Structured Query Language* (SQL) adalah sebuah bahasa yang dipergunakan untuk mengakses data dalam *database* relasional. Bahasa ini merupakan bahasa standar yang digunakan dalam manajemen *database* relasional. Saat ini hampir semua server *database* yang ada mendukung bahasa ini untuk melakukan manajemen datanya. Secara umum, SQL terdiri dari dua bahasa, yaitu *Data Definition Language* (DDL) dan *Data Manipulation Language* (DML). Implementasi DDL dan DML berbeda untuk tiap DBMS, namun secara umum implementasi tiap bahasa ini memiliki bentuk standar yang ditetapkan ANSI

### 2.7.1. Data Definition Language (DDL)

DDL digunakan untuk mendefinisikan, mengubah, serta menghapus *database* dan objek-objek yang diperlukan dalam *database*, misalnya tabel, view, *user*, dan sebagainya. Secara umum, DDL yang digunakan adalah `CREATE` untuk membuat objek baru, `USE` untuk menggunakan objek, `ALTER` untuk mengubah objek yang sudah ada, dan `DROP` untuk menghapus objek. DDL biasanya digunakan oleh administrator *database* dalam pembuatan sebuah aplikasi *database*.

### 2.7.2. *Data Manipulation Language (DML)*

DML digunakan untuk memanipulasi data yang ada dalam suatu tabel. Perintah yang umum dilakukan adalah:

- `SELECT` untuk menampilkan data
- `INSERT` untuk menambahkan data baru
- `UPDATE` untuk mengubah data yang sudah ada
- `DELETE` untuk menghapus data

## 2.8 *Delphi*

Delphi adalah kompiler/penerjemah bahasa Delphi (awalnya dari Pascal) yang merupakan bahasa tingkat tinggi sekelas dengan Basic atau C yang merupakan produk dari Borland corp. Bahasa Pemrograman di Delphi disebut bahasa procedural artinya bahasa/sintaknya mengikuti urutan tertentu/prosedur.

Delphi termasuk Keluarga Visual sekelas Visual Basic, Visual C, artinya perintah-perintah untuk membuat objek dapat dilakukan secara visual. Pemrogram tinggal memilih objek apa yang ingin dimasukkan kedalam Form/Window, lalu tingkah laku objek tersebut saat menerima event/aksi tinggal dibuat programnya.

Ada jenis pemrograman non-prosedural seperti pemrograman untuk kecerdasan buatan seperti bahasa Prolog. Delphi merupakan bahasa berorientasi objek, artinya nama objek, properti dan metode/procedure dikemas menjadi satu kemasan (encapsulate). Sebelum mempelajari ketiga struktur pemrograman ada baiknya kenali dahulu tampilan IDE, yang merupakan editor dan tools untuk membuat program Delphi.

Pada IDE akan ditampilkan Form baru yang merupakan aplikasi/program Window yang akan dibuat. Aplikasi/program berbasis windows sering disebut dengan jendela (window). Bagaimana membuat aplikasi berbasis windows (berbasis grafik dan



bukan berbasis teks seperti pada DOS)? Caranya dengan membuat sebuah form. Pada pemrograman berbasis windows, kita akan diperhadapkan pada satu atau beberapa jendela yang nampak dihadapan kita. Jendela ini dalam Delphi disebut juga dengan form. Pada pemrograman berbasis windows, kita akan diperhadapkan pada satu atau beberapa jendela yang nampak dihadapan kita. Jendela ini dalam Delphi disebut juga dengan form.

Delphi adalah sebuah perangkat lunak (bahasa pemrograman) untuk membuat program/aplikasi komputer berbasis windows.

Delphi merupakan bahasa pemrograman berbasis objek, artinya semua komponen yang ada merupakan objek-objek. Ciri sebuah objek adalah memiliki nama, properti dan method/procedure.

Delphi disebut juga visual programming artinya komponen-komponen yang ada tidak hanya berupa teks (yang sebenarnya program kecil) tetapi muncul berupa gambar-gambar.

Salah satu fasilitas Delphi yang yang paling banyak digunakan dan paling membantu pemrogram dalam menuliskan listing program adalah fasilitas source completion. Source completion adalah fasilitas yang disediakan Delphi yang berguna untuk melengkapi kode yang dituliskan pada kode editor secara otomatis. Dengan fasilitas ini proses menuliskan listing program akan menjadi lebih cepat. Tentu saja hal ini akan semakin menghemat waktu, biaya, dan tenaga.

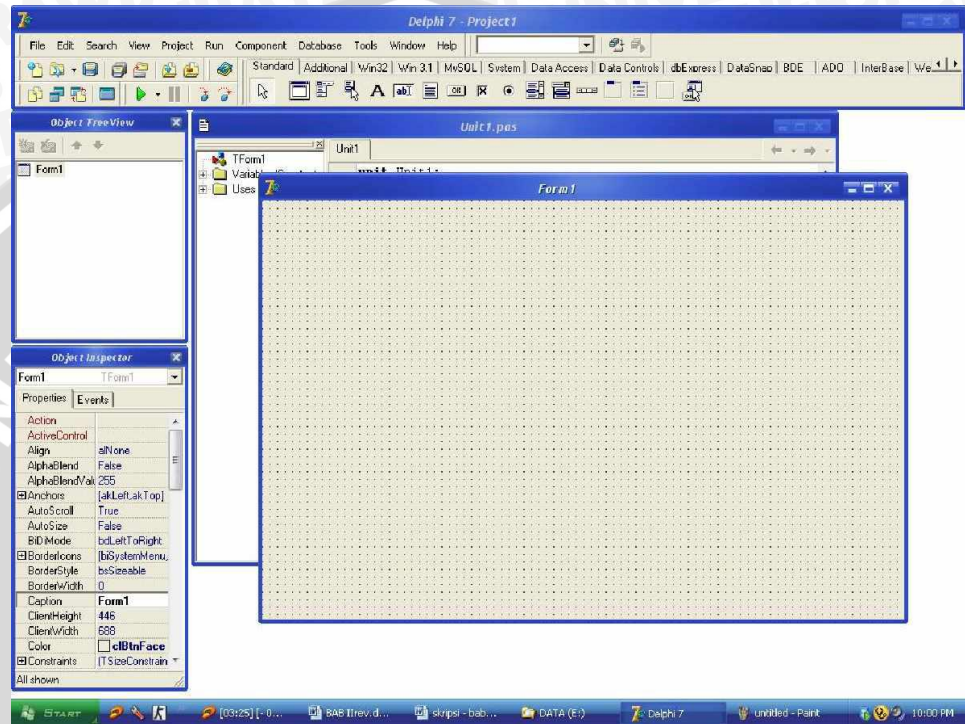
Khusus untuk pemrograman *database*, Delphi menyediakan object yang sangat kuat, canggih dan lengkap, sehingga memudahkan pemrograman dalam merancang, membuat dan menyelesaikan aplikasi *database* yang diinginkan. Selain itu juga Delphi dapat menangani data dalam berbagai format *database*, misalnya format MS-Acces, Sybase, Oracle, FoxPro, Informix, DB2 dan lain – lain. Format *database* yang dianggap asli dari Delphi adalah Paradox dan dBase.

Instalasi Delphi 7 pada lingkungan windows, seperti instalasi pada program-program aplikasi di lain di windows. Untuk lingkungan Linux, Borland menyediakan Borland Kylix yang merupakan versi Linux dari Borland Delphi.



### 2.8.1. Area Kerja Borland Delphi 7

Adapun area kerja Borland Delphi 7 adalah seperti gambar 2.1 di bawah ini:



Gambar 2.1 Area kerja delphi

Area kerja tersebut terdiri dari:

#### 1. Menu Utama

Menu utama terletak pada bagian paling atas dari area kerja. Menu utama antara lain terdiri dari submenu : File, Edit, Search, View, Project, Run dan seterusnya sampai submenu yang paling akhir yaitu: Help.



Gambar 2.2 Menu utama

#### 2. Toolbar

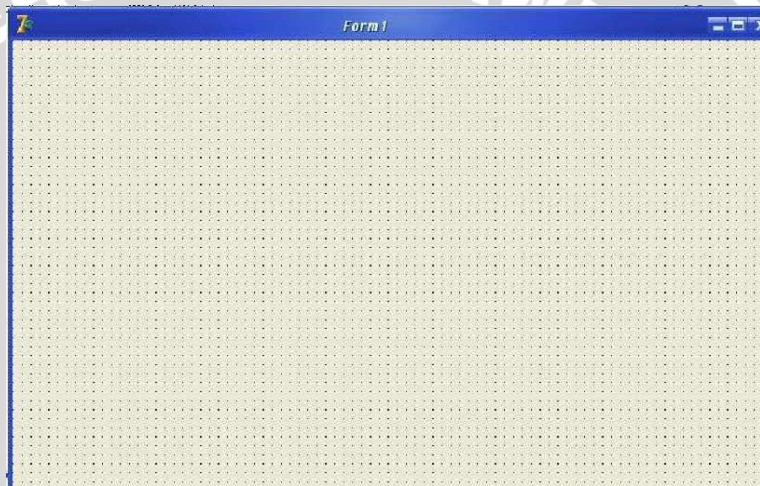
Pada menu toolbar terdiri dari ikon-ikon shortcut dari New Project, Save dan Run.



Gambar 2.3 Toolbar

### 3. Form Designer

*Form designer* merupakan area dimana akan dibuat project, di sini dapat diatur penempatan button-button, label, dan layout dari tampilan visual dari program yang akan dibuat.

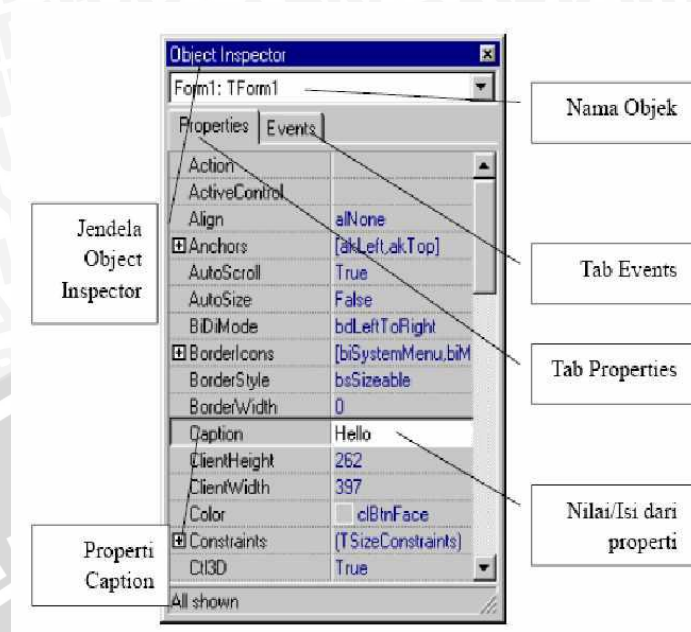


Gambar 2.4 Form designer

### 4. Object Inspector

Object Inspector berisi properti-properti dari form dan komponen/objek yang digunakan. Pada bagian ini terdapat peengaturan properti-properti dari form dan komponen/objek tersebut, antara lain: caption, ukuran, warna dan font pada form.





**Gambar 2.5** Object inspector

### 5. Component Palette

Semua komponen-komponen yang ada pada delphi atau yang terinstall pada delphi akan ditampilkan pada bagian component palette.



**Gambar 2.6** Component palette

#### 2.8.2. Form

Pada Delphi ada 3 buah file utama (\*.dpr, \*.pas dan \*.dfm). Masing-masing file tersebut mempunyai fungsi antara lain:

- a. \*.dpr adalah file proyek yang dibuat berisi program kecil untuk :
  - ❖ Mendefinisikan Unit yang ada dalam file proyek
  - ❖ Menginisialisasi data
  - ❖ Membangun form
  - ❖ Menjalankan aplikasi



**Contoh isi dpr :**

```
uses
Forms,
Unit1 in □Unit1.pas□ {Form1};
begin
Application.Initialize;
Application.CreateForm(Tform1, Form1);
Application.Run;
end.
```

- b. **\*.pas** adalah unit-unit (*pascal code file*), bisa terdiri satu atau banyak file
- c. **\*.dfm** adalah file definisi Form (*special pseudo code file*), bisa terdiri satu atau banyak file. Contohnya:

```
object Form1: Tform1
Left = 200
Top = 108
Width = 696
Height = 480
Caption = □Form1□
Font.Charset = DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = □MS Sans Serif□
Font.Style = []
PixelsPerInch = 96
TextHeight = 13
object Button1: Tbutton
Left = 176
Top = 116
Width = 75
Height = 25
Caption = □Button1□
TabOrder = 0
end
end
```

Setiap Form (.dfm) harus memiliki sebuah Unit (.pas),  
tetapi dapat memiliki Unit tanpa sebuah Form (hanya kode saja).



## BAB III

### METODE PENELITIAN

Adapun metode penelitian yang digunakan pada penyusunan skripsi ini adalah :

#### 3.1 Studi Literatur

Studi literatur berguna untuk memperoleh data dan menjelaskan dasar teori yang digunakan untuk menunjang penulisan skripsi. Teori-teori pendukung tersebut meliputi:

1. Mempelajari buku-buku yang menjelaskan tentang *MySQL* dan *Delphi*.
2. Mempelajari metode pencarian data, pencocokan data, pengambilan data dan pengolahan data pada *database*
3. Mempelajari teknik – teknik yang ada pada *MySQL*
4. Mempelajari teknik – teknik yang ada pada *Borland Delphi 7.0*

#### 3.2 Analisis Kebutuhan

Analisis kebutuhan bertujuan untuk mendapatkan semua kebutuhan yang diperlukan dalam pembuatan aplikasi penerjemah.

#### 3.3 Perancangan

Perancangan aplikasi penerjemah dilakukan setelah semua kebutuhan sistem didapatkan melalui tahap analisis kebutuhan. Perancangan aplikasi penerjemah diawali dengan melakukan desain *database* dari aplikasi penerjemah itu sendiri. Kemudian akan dilakukan perancangan sistem penerjemah itu sendiri berdasarkan *database* yang telah dirancang sebelumnya. Perancangan *database* dilakukan dengan menggunakan ERD(*Entity Relationship Diagram*) sedangkan perancangan aplikasi menggunakan *flowchart* untuk menjelaskan garis besar sistem dan algoritma yang digunakan dalam proses pembuatan aplikasi.



### 3.4 Implementasi

Implementasi dilakukan dengan mengacu kepada perancangan perangkat lunak. Implementasi perangkat lunak dilakukan dengan menggunakan bahasa pemrograman *delphi* dan menggunakan DBMS *MySQL*. Sistem dapat mengalami perubahan dari rancangan sebelumnya pada tahapan ini, bila ternyata diperlukan pergantian.

### 3.5 Pengujian Sistem

Pengujian dilakukan untuk menjamin dan memastikan bahwa sistem yang telah dirancang memiliki tingkat kesalahan yang kecil dan memiliki akurasi penerjemahan kata yang benar. Pengujian yang dilakukan meliputi kecepatan menerjemahkan kalimat dan tingkat kesesuaian makna dari kalimat hasil terjemahan.

### 3.6 Kesimpulan dan Saran

Pada tahap ini, diambil dari hasil pengujian dan analisa terhadap aplikasi yang telah dibuat. Tahap Selanjutnya adalah membuat saran untuk perbaikan terhadap penelitian selanjutnya sehingga dapat menyempurnakan kekurangan-kekurangan yang ada dan mengembangkan hasil yang diperoleh dari skripsi ini.

## BAB IV

### PERANCANGAN DAN IMPLEMENTASI

*Software* yang dibuat dalam tugas akhir ini adalah *software* penerjemah yang digunakan untuk mengubah *input* berupa *text* kalimat berbahasa Indonesia menjadi *output* berupa *text* kalimat berbahasa Inggris .

Bab ini membahas perancangan dan implementasi aplikasi penerjemah Bahasa Inggris ke Bahasa Indonesia yang dikerjakan dengan beberapa tahap meliputi: membangun sebuah *database* yang digunakan sebagai dasar dari aplikasi itu sendiri, pembuatan program penerjemah berdasarkan *database* yang telah dibangun, proses pengkoneksian antara *database* dan program penerjemah, pembuatan *User Interface* dari aplikasi penerjemah. Pembuatan secara bertahap tersebut agar memudahkan penganalisaan aplikasi pada setiap bagian maupun aplikasi secara keseluruhan.

#### 4.1. Kebutuhan Sistem

Perangkat lunak ini menggunakan *windows* XP sebagai sistem operasi utamanya. Kebutuhan sistem itu sendiri dapat dibedakan menjadi dua, yaitu kebutuhan perangkat lunak dan perangkat keras.

##### 4.1.1. Perangkat Keras

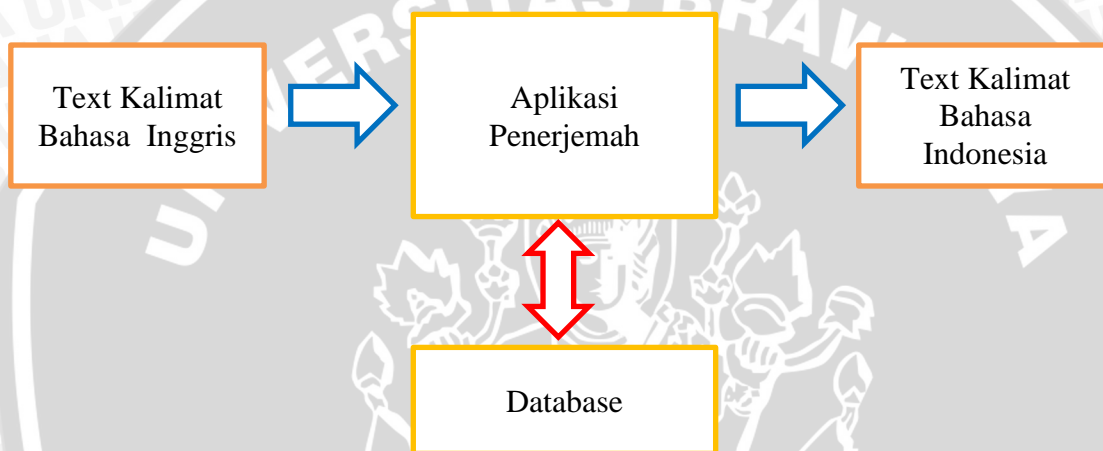
Untuk perangkat kerasnya, membutuhkan komputer dengan spesifikasi minimum pentium IV dengan RAM 512 Mbytes. Penulis mengaplikasikan perangkat lunak ini pada sebuah notebook Intel Pentium i7 2,67 GHz dengan RAM 4096 Mbytes.

##### 4.1.2. Perangkat Lunak

Untuk perangkat lunak yang digunakan dalam pembuatan aplikasi ini ialah sistem operasi *windows* XP. *phpMyAdmin Database Manager Version 2.9.0. 2* dan *Borland Delphi 7*

## 4.2. Perancangan Aplikasi Secara Umum

Aplikasi ini dapat menerima *input* berupa *text* kalimat yang menggubakan Bahasa Inggris, kemudian kalimat tersebut di proses didalam sistem yang terkoneksi dengan *database* sehingga dapat dihasilkan *output* berupa *text* kalimat dengan menggunakan Bahasa Indonesia dengan makna yang seakurat mungkin dengan makna kalimat aslinya. Berikut gambar dari diagram aplikasi tersebut.



Gambar 4.1 Diagram Sistem

## 4.3. Pembuatan Aplikasi Penerjemah

Dalam pembuatan aplikasi ini terdapat tiga komponen, yang pertama adalah *database* dari sistem itu sendiri kemudian *form* yang dibuat menggunakan *delphi* yang merupakan GUI (*Graphical User Interface*) dari aplikasi penerjemah itu sendiri. Dan terakhir, sebuah form yang akan terhubung dengan *database* sistem. Cara kerja dari aplikasi ini sendiri dapat dijelaskan sebagai berikut:

### 4.3.1. Database sistem

Agar aplikasi dapat berjalan sesuai yang diinginkan maka *databasenya* juga harus mendukung, oleh karena itu *database* sebaiknya



dibuat dari awal. Karena harus mendukung frasa, maka dalam *database* harus ada tabel yang berisi frasa. Maka dibuat **tabel frasa**.

Dalam Bahasa Inggris terdapat kata-kata yang harus ada sebagai pelengkap atau penghubung dalam tata bahasa-nya, oleh karena itu disediakan tabel untuk kata- yang seperti ini. Maka dibuat **tabel disjunct**.

Kemudian untuk kata-kata dasar, baik yang beraturan maupun tidak beraturan juga harus ada yang mengakomodir, oleh karena itu dibuat **tabel kamus**.

Lalu terdapat juga **tabel arti** tabel ini berguna jika terdapat kata yang bermakna ganda, maka akan terdapat dua kalimat *output* yang berbeda, dimana kalimat tersebut ada yang benar dan yang salah. Dengan adanya tabel ini *user* dapat menyimpan kalimat yang maknanya benar.

Kemudian yang terakhir terdapat **tabel kalimat temporary** untuk menyimpan kalimat hasil olahan program sebelum dikeluarkan sebagai kalimat *output*.

#### ❖ Tabel Kamus

	Field	Jenis	Penyortiran	Atribut	Kosong	Default
<input type="checkbox"/>	<b>english</b>	varchar(20)	utf8_general_ci		Tidak	
<input type="checkbox"/>	<b>indonesia</b>	varchar(20)	utf8_general_ci		Ya	NULL
<input type="checkbox"/>	<b>jenis</b>	varchar(20)	utf8_general_ci		Ya	NULL
<input type="checkbox"/>	<b>ket_jenis</b>	varchar(20)	utf8_general_ci		Ya	NULL
<input type="checkbox"/>	<b>indonesia2</b>	char(20)	utf8_general_ci		Tidak	

Gambar 4.2 Tabel Kamus

Tabel 4.1 Keterangan *field* tabel kamus

Field	Keterangan
english	Kata dalam bhs. Inggris
indonesia	Terjemahan kata dalam bhs. indonesia
Indonesia2	Terjemahan alternatif kata dalam bhs. indonesia
Jenis	Jenis kata

Tabel ini berisi kata dasar dalam Bahasa Inggris dan terjemahannya dalam Bahasa Indonesia. Setiap kata dasar dilengkapi keterangan jenis kata, apakah kata tersebut termasuk *verb*, *noun*, atau *adjective* dan lain sebagainya seperti yang telah dijelaskan pada dasar teori di bab sebelumnya.

❖ Tabel Kalimat *Temporary*

	Field	Jenis	Penyortiran	Atribut	Kosong	Default
<input type="checkbox"/>	<u>nomer</u>	int(10)			Tidak	
<input type="checkbox"/>	english	varchar(20)	utf8_general_ci		Ya	NULL
<input type="checkbox"/>	indonesia	varchar(30)	utf8_general_ci		Ya	NULL
<input type="checkbox"/>	jenis	varchar(20)	utf8_general_ci		Ya	NULL

Gambar 4.3 Tabel Kalimat *Temporary*

Tabel 4.2 Keterangan *field* tabel kalimat *temporary*

Field	Keterangan
nomer	Kata ke- dalam kalimat
english	Kata dalam bhs. Inggris
indonesia	Terjemahan kata dalam bhs. indonesia
Jenis	Jenis kata

Tabel ini digunakan sebagai penyimpan kata yang telah dimasukkan ke dalam program yang nantinya akan diterjemahkan

❖ Tabel *Disjunct*

	Field	Jenis	Penyortiran	Atribut	Kosong	Default
<input type="checkbox"/>	<b>english</b>	varchar(20)	utf8_general_ci		Tidak	
<input type="checkbox"/>	<b>indonesia</b>	varchar(20)	utf8_general_ci		Ya	NULL
<input type="checkbox"/>	<b>jenis</b>	varchar(20)	utf8_general_ci		Ya	NULL

Gambar 4.4 Tabel *Disjunct*

Tabel 4.3 Keterangan *field* tabel *Disjunct*

Field	Keterangan
english	Kata dalam bhs. Inggris
indonesia	Terjemahan kata dalam bhs. indonesia
Jenis	Jenis kata

Tabel ini berisi kata-kata yang dibutuhkan dalam aturan-aturan atau tata bahasa yang berlaku dalam Bahasa Inggris untuk menentukan bagaimana susunan kata-kata yang ada sehingga dapat diterjemahkan ke Bahasa Indonesia seakurat mungkin.

❖ Tabel Frasa

	Field	Jenis	Penyortiran	Atribut	Kosong	Default
<input type="checkbox"/>	<b>english</b>	varchar(20)	utf8_general_ci		Tidak	
<input type="checkbox"/>	<b>indonesia</b>	varchar(30)	utf8_general_ci		Ya	NULL

Gambar 4.5 Tabel *Disjunct*

Tabel 4.4 Keterangan *field* tabel Frasa

Field	Keterangan
english	Kata dalam bhs. Inggris
indonesia	Terjemahan kata dalam bhs. indonesia

Tabel ini berisi frasa atau idiom yang ada dalam Bahasa Inggris.



❖ Tabel arti

	Field	Jenis	Penyortiran	Atribut	Kosong	Default
<input type="checkbox"/>	no	int(11)			Tidak	
<input type="checkbox"/>	eng	char(255)	utf8_general_ci		Tidak	
<input type="checkbox"/>	ind	char(255)	utf8_general_ci		Tidak	

Gambar 4.6 Tabel arti

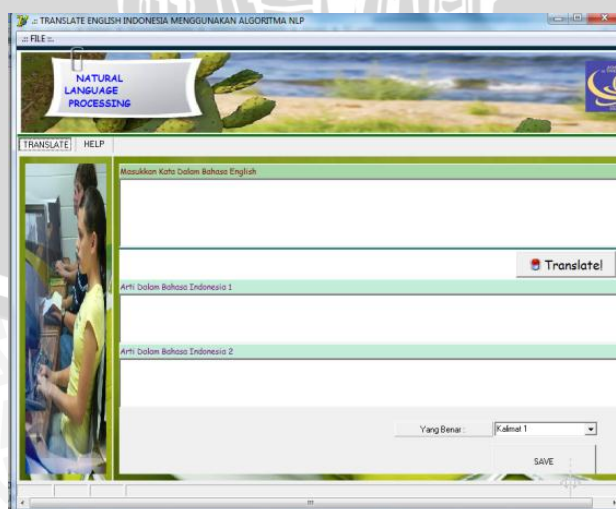
Tabel 4.5 Keterangan *field* tabel arti

Field	Keterangan
no	Memberi nomor pada kalimat
Eng	Kalimat Bahasa Inggris yang pernah dimasukkan
ind	Terjemahan dari kalimat Bahasa Inggris yang benar

Tabel ini digunakan untuk menyimpan kata yang pernah dimasukkan dan terjemahannya yang benar.

**4.3.2. GUI(Grahical User Interface) sistem**

GUI(*Graphical User Interfacce*) merupakan antar muka antar muka untuk memudahkan *user* dalam menggunakan aplikasi yang dibuat, GUI aplikasi ini dapat dilihat pada gambar berikut:



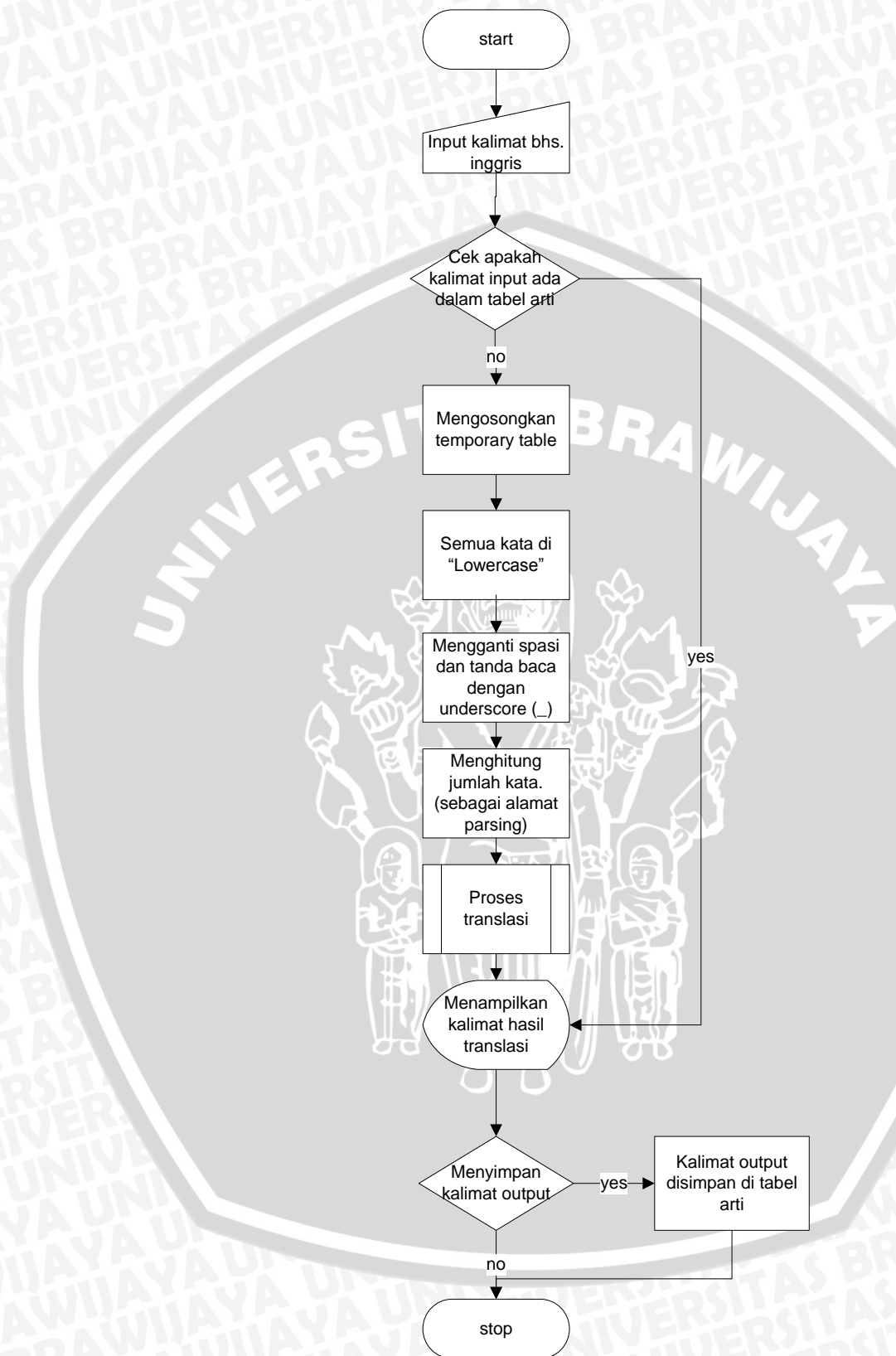
Gambar 4.6 Tampilan GUI Aplikasi Penerjemah

### 4.3.3. Cara Kerja Aplikasi Penerjemah

Pertama-tama, *textfield input* pada aplikasi penerjemah diisi berupa kalimat berbahasa Inggris. Kemudian dilakukan pengecekan pada tabel arti, apakah kalimat yang telah dimasukkan pernah diterjemahkan sebelumnya, jika ditemukan maka hasil terjemahannya yang telah disimpan langsung dikeluarkan. Jika tidak ditemukan maka *Input* kalimat diambil kemudian di kecilkan hurufnya. Lalu tabel kalimat *temporary* dikosongkan. kemudian kata-kata yang ada dipisahkan satu-persatu dan dihitung jumlah kata yang ada. Kemudian mulai proses translasi yang akan dijelaskan pada sub bab selanjutnya. Setelah proses translasi selesai hasil translasi ditampilkan dalam *textfield output*. Setelah kalimat keluar akan terdapat dua *output* jika ada kata yang bermakna ganda, maka *output* akan berbeda, dan *user* dapat memilih terjemahan kalimat mana yang benar. Selanjutnya jika kalimat tersebut dimasukkan lagi, maka terjemahan kalimat yang telah disimpan tersebut dikeluarkan.

Untuk lebih jelasnya sistem kerja aplikasi penerjemah dapat dilihat dalam diagram alir berikut:



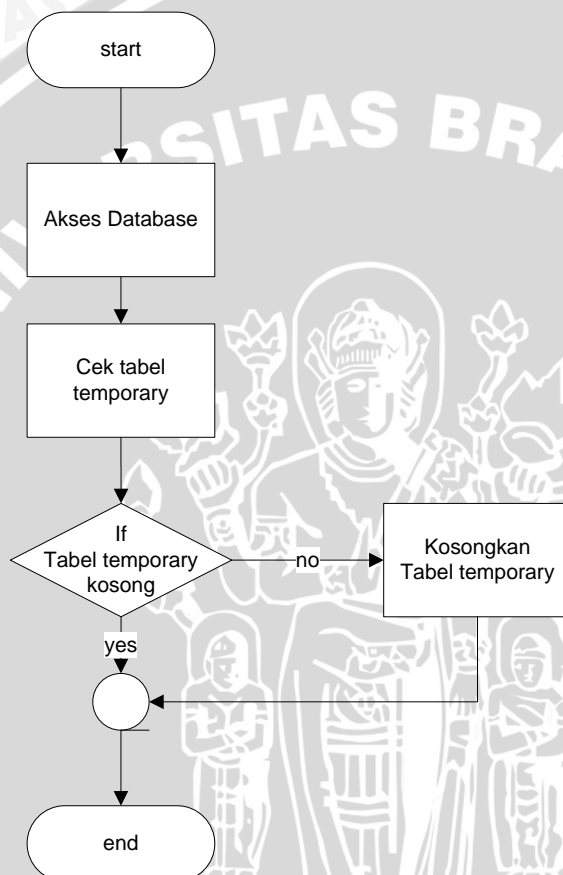


Gambar 4.7 flowchart Cara Kerja Aplikasi Penerjemah



Keterangan:

1. *Input* Kalimat Bahasa Inggris. *User* memasukkan kalimat Bahasa Inggris ke ngdalam *textfield* yang telah disediakan
2. Cek apakah kalimat *input* sudah pernah diterjemahkan, bila iya maka langsung menampilkan kalimat hasil translasi. Bila tidak maka lanjut ke proses setelahnya.
3. Mengosongkan Tabel Kalimat *Temporary*.

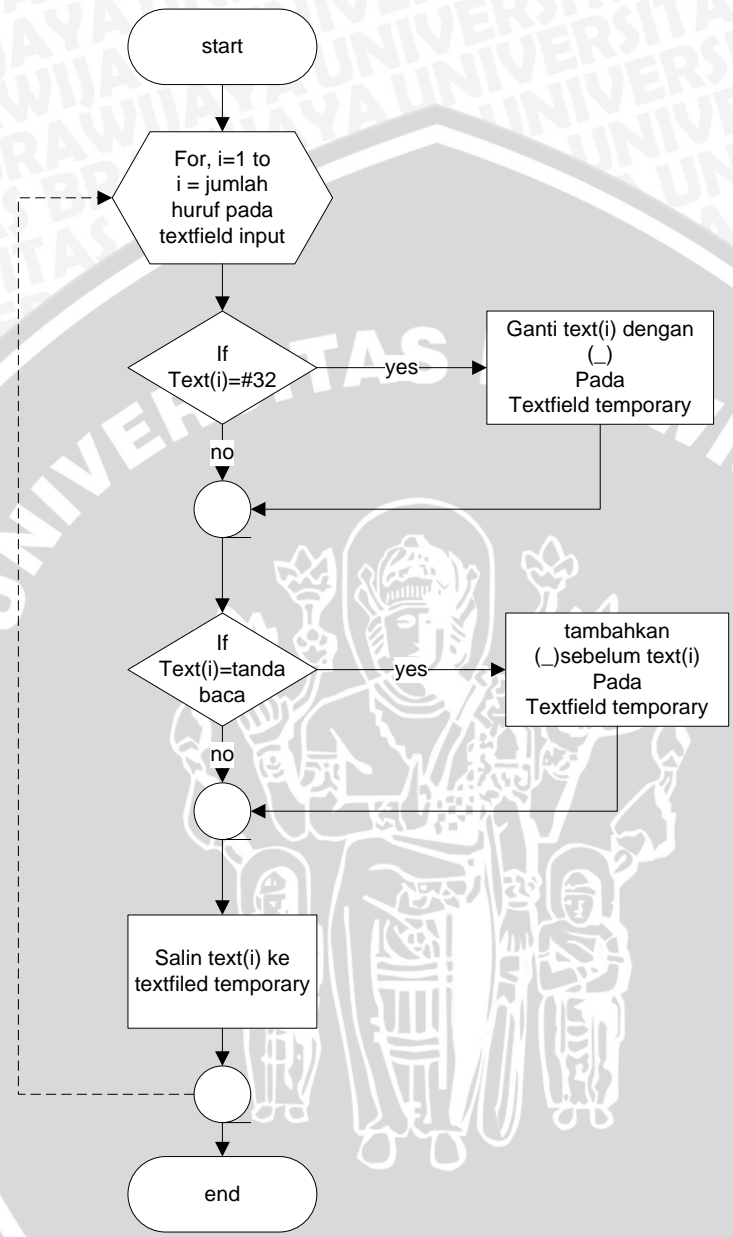


Gambar 4.8 *flowchart* metode pengosongan table *temporary*

Hal ini bertujuan agar tabel *temporary* untuk menyimpan kalimat sementara yang berguna dalam pencocokan kata dalam *database* menjadi kosong dan siap untuk diisi kata-kata yang baru.

4. Kata-kata yang ada pada *textfield* dijadikan huruf kecil(*lowercase*). dimaksudkan agar tidak terjadi kesalahan pada saat pencocokan kata di dalam *database*, karena kata-kata yang terdapat didalam *database* semuanya *lowercase*

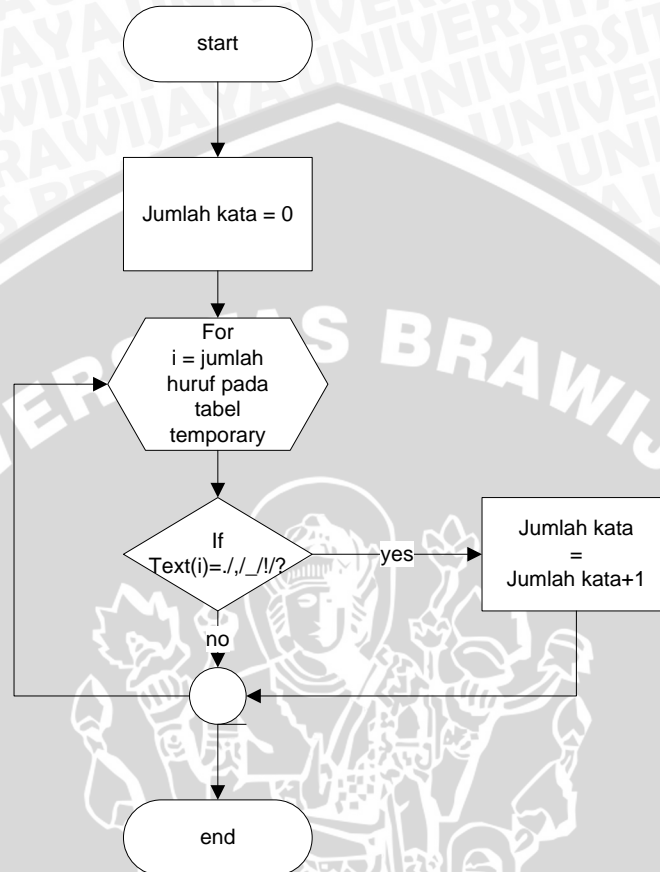
5. Mengganti spasi dan tanda baca dengan *underscore*(\_)



Gambar 4.9 flowchart metode penganti spasi

kegunaannya agar tidak terjadi kesalahan dalam pembacaan kata2 yang dan tanda baca dapat dikenali sebagai satu kata agar nantinya bias di keluarkan lagi sebagai *output*.

## 6. Menghitung jumlah Kata

Gambar 4.10 *flowchart* metode penghitung jumlah kata

Penghitungan jumlah kata ini berguna sebagai alamat parsing yang akan digunakan untuk dalam proses translasi.

7. Proses translasi dari kata yang telah di *input* pada *textfield* dan di optimasi melalui proses-proses setelahnya. Untuk lebih lanjut mengenai proses translasi akan dijelaskan pada sub bab selanjutnya.

8. Kalimat hasil dari proses translasi ditampilkan pada *output textfield*. Disini jika terdapat kata yang bermakna ganda, maka akan terdapat dua terjemahan yang berbeda. Terjemahan tersebut *pasti* ada yang benar dan yang salah, diharapkan *user* menyimpan hasil yang benar,



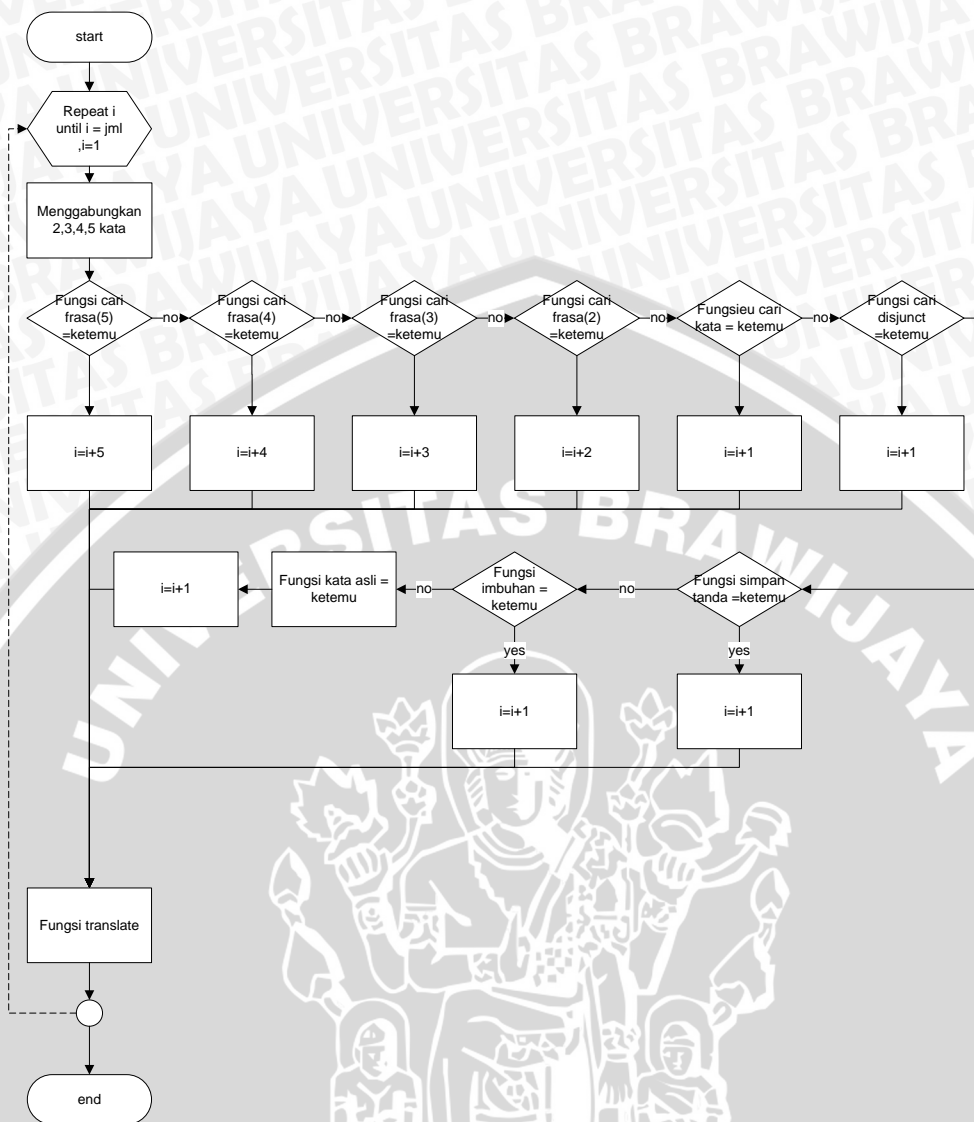
agar ketika kalimat yang sama dimasukkan lagi maka aplikasi akan langsung menerjemahkannya dengan benar.

9. *User* dapat menyimpan kalimat hasil terjemahan. Jika *user* menyimpan kalimat terjemahan, maka kalimat tersebut akan disimpan pada tabel arti.

#### 4.3.4. Proses Translasi

Dalam Proses Translasi terdapat beberapa fungsi utama dimana penggunaan fungsi itu sendiri tergantung dari kalimat *input*. Langkah awal dalam proses translasi ini adalah melakukan pengecekan pada kalimat mulai awal hingga akhir. Dari proses pengecekan tersebut akan diketahui fungsi mana yang harus digunakan untuk memproses tiap kata dalam kalimat tersebut. Perlakuan atau fungsi yang digunakan pada tiap kata dapat berbeda antara satu dengan yang lainnya.

Fungsi yang terdapat dalam Proses Translasi ini diantaranya adalah fungsi untuk mencari frasa, yang akan digunakan jika terdapat frasa dalam kalimat *input*. Kemudian terdapat juga fungsi untuk mengekstraksi imbuhan dari suatu kata, hal ini di perlukan karena dalam Bahasa Inggris imbuhan berperan penting dalam *grammar* atau tata-bahasa. Salah satunya, imbuhan menentukan apakah kejadian yang diungkapkan dalam kalimat yang di masukkan telah terjadi atau sedang terjadi. Fungsi yang ketiga adalah fungsi untuk mencari kata tertentu yang bersangkutan dengan tata bahasa yang ada dalam Bahasa Inggris. Kemudian terdapat juga fungsi untuk menyimpan tanda baca. Fungsi selanjutnya adalah fungsi untuk mencocokkan kata dasar yang ada dalam kalimat *input* dengan kata dasar dan terjemahannya yang ada dalam *database*. Fungsi yang terakhir adalah untuk mendeteksi jenis kata yang ada dan memberi perlakuan sesuai tata bahasa yang ada. Lebih lanjut mengenai bagaimana proses translasi dapat dijelaskan sebagai berikut:



Gambar 4.11 flowchart proses penelusuran kalimat input

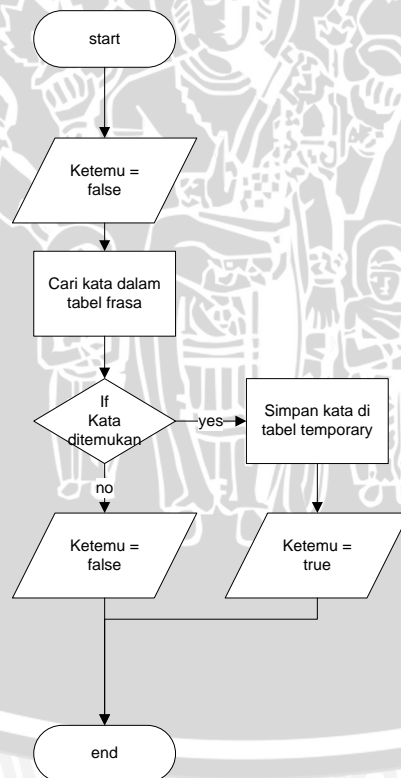
Proses ini akan menelusuri kalimat mulai dari kata yang pertama, kemudian melakukan penggabungan kata sebanyak 2 kata sampai dengan 5 kata. Setelah proses tersebut dilakukan pengecekan melalui fungsi-fungsi yang ada, kemudian melangsi proses dari awal pada kata kedua sampai pada kata terakhir sesuai dengan jumlah kata yang telah diketahui pada proses sebelumnya.

Untuk lebih lanjut mengenai fungsi yang terdapat dalam aplikasi penerjemah ini akan dijelaskan sebagai berikut:



### 1. Fungsi cari frasa

Fungsi ini digunakan untuk mencari frasa yang terdapat pada kalimat. Fungsi ini dijalankan setelah menggabungkan kata-kata yang ada dalam kalimat *input*, pertama-tama fungsi ini akan melakukan pengecekan apakah ada frasa pada 5 kata yang telah digabung dalam tabel frasa, kemudian melakukan hal yang sama pada gabungan 4 kata, 3 kata, dan 2 kata. Jika ada frasa yang ditemukan maka akan disimpan dalam tabel *temporary* dan nilai keluaran dari fungsi ini yang di representasikan dalam variabel “ketemu” akan bernilai *true* dan *counter* kata akan bertambah sesuai dengan jumlah kata yang ditemukan. Jika tidak ada frasa yang ditemukan maka nilai keluaran akan *false*. Lebih jelasnya dapat dilihat pada diagram alir berikut:

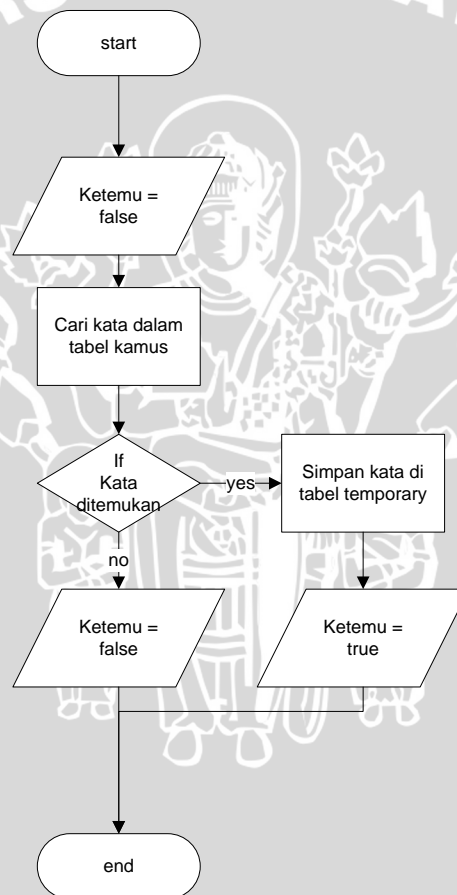


Gambar 4.12 *flowchart* fungsi cari frasa



## 2. Fungsi Cari kata

Fungsi ini digunakan untuk mencari kata yang ada dalam tabel kamus. Fungsi ini dijalankan jika tidak ditemukan adanya frasa. Fungsi ini mengecek apakah kata yang di masukkan ada didalam tabel kamus. Jika ada maka kata tersebut akan disimpan dalam tabel *temporary*. dan nilai keluaran dari fungsi ini akan bernilai *true*. Kemudian counter kata akan bertambah 1. Jika ternyata tidak ditemukan maka nilai keluaran akan bernilai *false*. Dan akan menjalankan fungsi selanjutnya. Untuk lebih jelasnya dapat dilihat dalam diagram berikut:

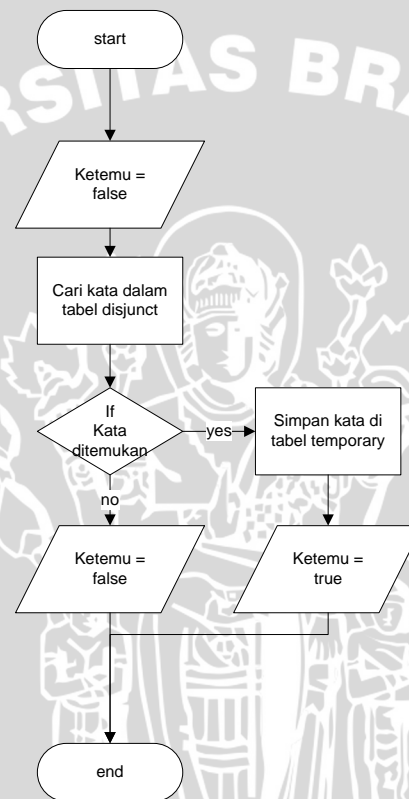


Gambar 4.13 *flowchart* fungsi cari kata

## 3. Fungsi cari *disjunct*

Fungsi ini digunakan untuk mencari kata yang ada dalam tabel *disjunct*. Fungsi ini dijalankan jika tidak ditemukan

adanya kata dalam tabel kamus. Fungsi ini mengecek apakah kata yang di masukkan ada didalam tabel disjunct. Jika ada maka kata tersebut akan disimpan dalam tabel *temporary* dan nilai keluaran dari fungsi ini akan bernilai *true*. Kemudian *counter* kata akan bertambah 1. Jika ternyata tidak ditemukan maka nilai keluaran akan bernilai *false*. Dan akan menjalankan fungsi selanjutnya. Untuk lebih jelasnya dapat dilihat dalam diagram berikut:



Gambar 4.14 *flowchart* fungsi cari disjunct

#### 4. Fungsi Simpan tanda

Fungsi ini digunakan untuk mengecek apakah kata yang ada merupakan tanda baca. Jika kata yang da adalah tanda baca maka akan langsung disimpan dalam tabel *temporary*, dan nilai keluaran akan bernilai *true*. . Kemudian *counter* kata akan bertambah 1. Jika tidak, maka akan lanjut ke fungsi

selanjutnya dan nilai keluaran akan bernilai *false*. Untuk lebih jelasnya dapat dilihat dalam diagram berikut:



Gambar 4.15 *flowchart* fungsi *simpan tanda*

#### 5. Fungsi Imbuhan

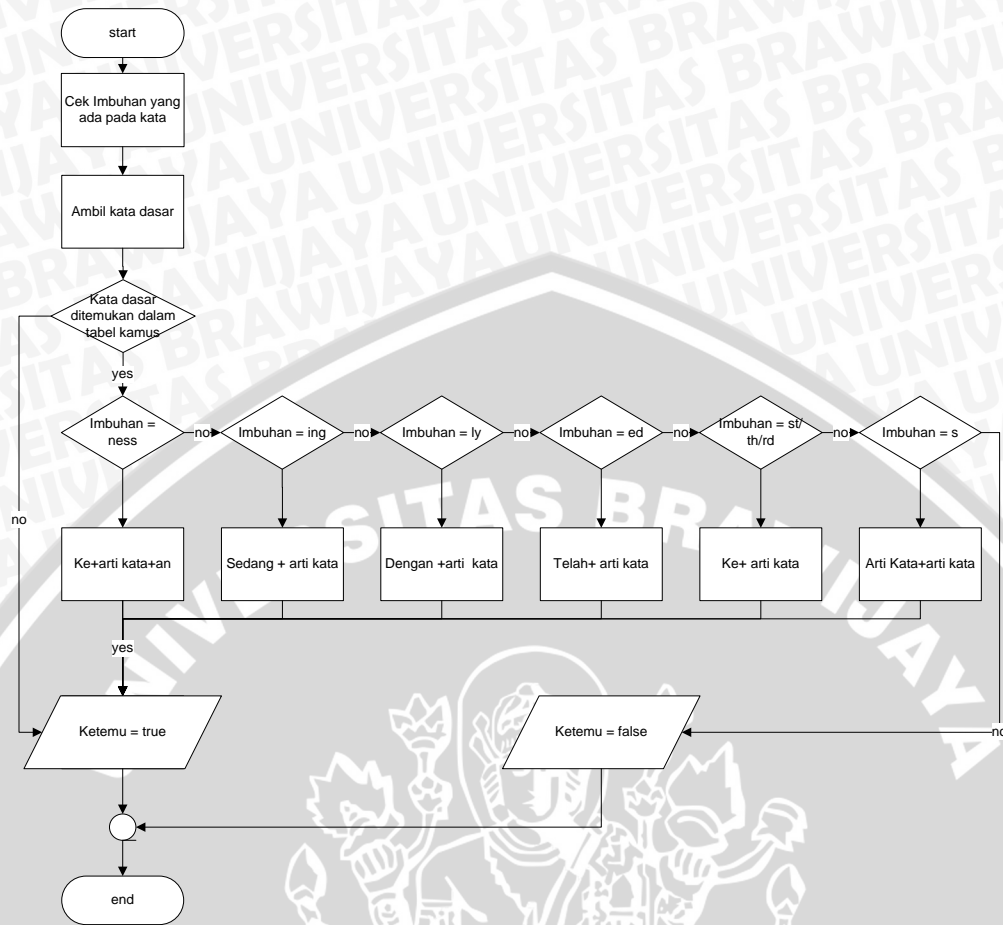
Fungsi ini dipergunakan untuk mengecek ada atau tidaknya imbuhan pada kata. Prosedur pengecekan imbuhan tersebut adalah sebagai berikut:

- Dilakukan pengecekan 1 sampai 4 huruf terakhir pada kata.
- Huruf-huruf tersebut kemudian dipisahkan, atau dengan kata lain dilakukan ekstraksi kata dasar.
- Kata dasar yang telah diambil di cek pada tabel kamus.
- Jika kata dasar tersebut tidak ada, maka lanjut ke fungsi selanjutnya, jika ternyata terdapat dalam tabel kamus, maka akan dilakukan pengecekan terhadap imbuhan.



- Jika imbuhan yang ditemukan adalah “*ness*”, maka terjemahan dari kata tersebut akan ditambahkan imbuhan ke – an.
- Jika imbuhan yang ditemukan adalah “*ing*”, maka dapat dipastikan kata ini menandakan bahwa kalimat merupakan kalimat *progressive* atau *continuous* oleh karena itu terjemahan dari kata tersebut akan ditambahkan kata “sedang” sebelumnya.
- Jika imbuhan yang ditemukan ialah “*ly*”, ini berarti menandakan kata tersebut menerangkan cara, oleh karenanya terjemahan kata akan ditambahkan kata “dengan” sebelumnya.
- Jika imbuhan yang dideteksi adalah “*ed*”, maka hal ini mengindikasikan bahwa kalimat adalah kalimat lampau atau *past tense* oleh karena itu sebelum kata terjemahan akan ditambahkan kata “telah”.
- Jika imbuhan yang ditemukan adalah “*s*”, berarti kata tersebut bersifat jamak atau *plural*. Dengan demikian maka terjemahan kata tersebut akan diulang sekali.

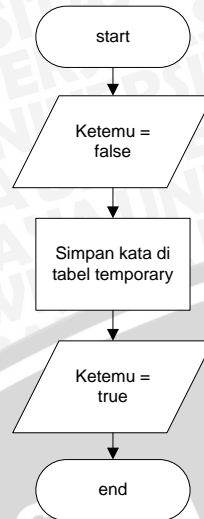
Setelah kata terjemahan diberikan perlakuan sesuai imbuhan yang ada maka kata tersebut akan disimpan dalam tabel *temporary* dan variabel “ketemu” akan bernilai *true*. Kemudian *counter* kata akan bertambah 1. Sebaliknya jika tidak ditemukan imbuhan maka variabel “ketemu” akan bernilai *false* dan lanjut ke fungsi selanjutnya. Agar lebih jelas dapat dilihat pada diagram dibawah ini:



Gambar 4.16 flowchart fungsi Imbuhan

## 6. Fungsi kata Asli

Jika fungsi ini dilajankan maka kata yang dimasukkan tidak ditemukan dimanapun dan kata tersebut akan langsung disimpan dalam tabel *temporary* dan nilai keluaran yang direpresentasikan dalam variabel *ketemu* akan bernilai *true*, *counter* kata akan bertambah 1. Kemudian akan langsung menjalankan fungsi selanjutnya. Diagram dibawah ini akan merepresentasikan bagaimana fungsi ini bekerja.

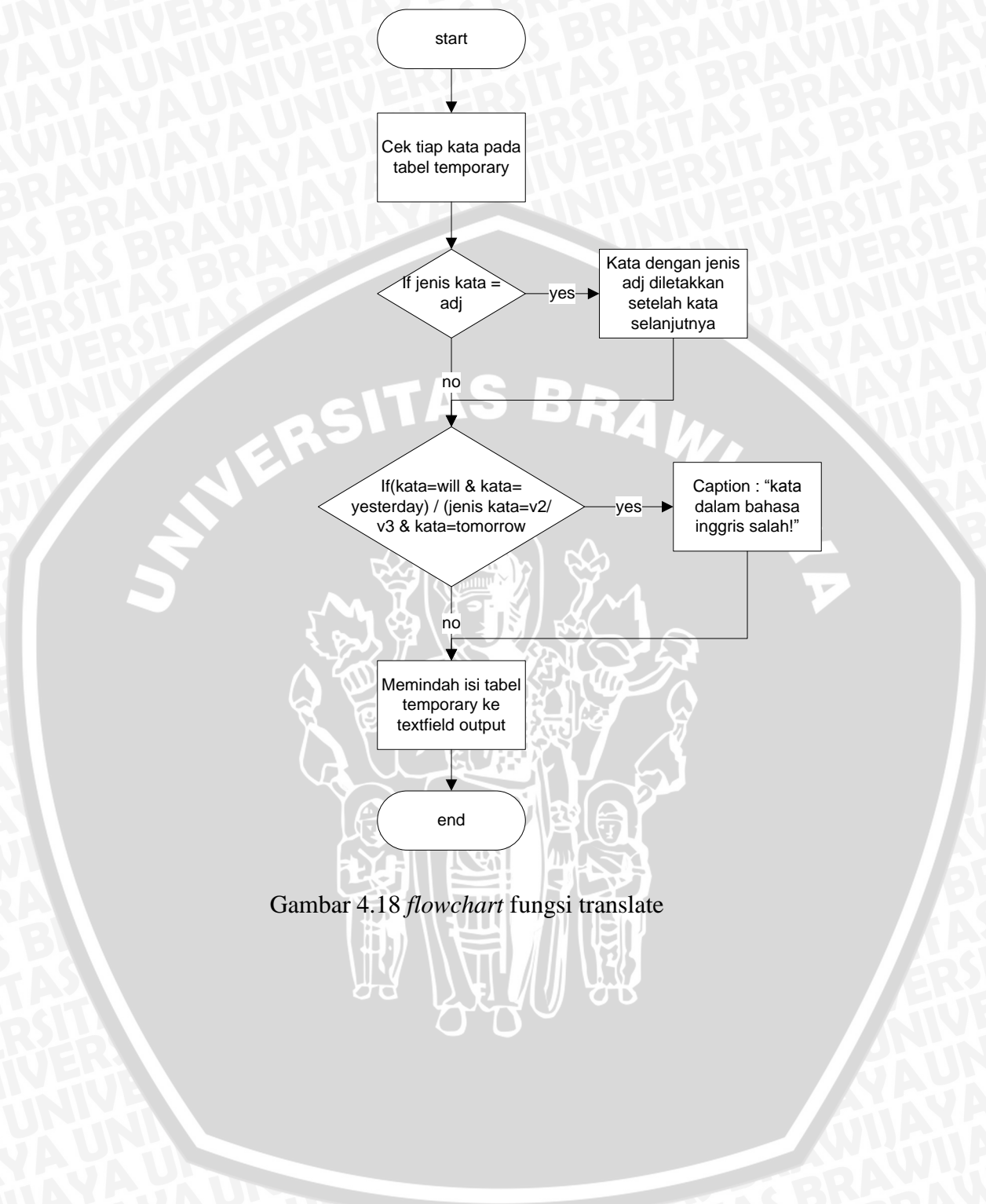


Gambar 4.17 flowchart fungsi kata asli

#### 7. Fungsi translate

Fungsi ini mempunyai dua kegunaan utama, yang pertama, fungsi ini akan mengecek jenis kata pada kata-kata yang telah dimasukkan ke dalam tabel *temporary* oleh fungsi-fungsi sebelumnya. Bila terdapat kata *will* dan kata *yesterday* maka akan ada *caption* yang menandakan bahwa kalimat *input* salah. *Caption* tersebut juga muncul apabila dideteksi ada kata lampau dan kata *tomorrow*. Jika terdapat kata yang berjenis kata *adjective* maka kata tersebut akan diletakkan setelah kata selanjutnya. Misalnya: kata ke-2 dalam tabel adalah *black* yang merupakan *adjective* dan kata ke-3 dalam tabel adalah *cat*, maka kata ke-3 dan kata ke-2 dibalik posisinya sehingga kata ke-2 menjadi *cat* dan kata ke-3 menjadi *black*. Hal ini dimaksudkan agar hasil terjemahan sesuai dengan kaidah tata bahasa yang berlaku. Kemudian kegunaan yang kedua adalah untuk memindahkan isi terjemahan Bahasa Indonesia dari tabel *temporary* ke *textfield output*. Diagram dibawah ini akan merepresentasikan bagaimana fungsi ini bekerja.





Gambar 4.18 flowchart fungsi translate

## BAB V

### PENGUJIAN SISTEM

Pengujian dan analisis dilakukan untuk mengetahui apakah sistem telah bekerja sesuai perancangan

#### 5.1 Metode Pengujian

Metode pengujian yang digunakan adalah dengan menguji aplikasi untuk menerjemahkan 5 kalimat pada tiap *main tenses* (kalimat-kalimat utama) yang terdapat dalam Bahasa Inggris. Yaitu:

- *Past simple*
- *Past progressive*
- *Past perfect*
- *Past perfect progressive*
- *Present simple*
- *Present progressive*
- *Present perfect*
- *Present perfect progressive*
- *Future simple*
- *Future progressive*
- *Future perfect*
- *Future perfect progressive*

Sumber dari kalimat tersebut diambil dari buku *Understanding english grammar* (betty Schramfer Azar:1993) dan Swarabhaskara.com

#### 5.2 Hasil Pengujian

Berikut ini merupakan hasil pengujian dari aplikasi penerjemah berdasarkan jenis kalimatnya:

5.2.1. *Past Simple*Tabel 5.1 Hasil pengujian kalimat *past simple*

English	Indonesia	Keterangan
<i>i walked to school yesterday</i>	saya berjalan ke sekolah kemarin	Berhasil diterjemahkan
<i>he lived in paris for ten years, but now he is living in rome</i>	dia(lelaki) telah hidup di paris untuk/selama sepuluh tahun-tahun, tetapi sekarang dia(lelaki) sedang hidup di rome.	Berhasil diterjemahkan
<i>i bought a new car three days ago</i>	saya telah membeli sebuah mobil baru tiga hari yang lalu	Berhasil diterjemahkan
<i>when he heard a strange noise, he got up to investigate</i>	kapan/ketika dia(lelaki) telah mendengar sebuah kebisingan aneh, dia(lelaki) telah bangun ke/untuk menyelidiki.	Berhasil diterjemahkan
<i>it rained this morning</i>	telah hujan pagi ini	Berhasil diterjemahkan

5.2.2. *Past Progressive*Tabel 5.2 Hasil pengujian kalimat *past progressive*

English	Indonesia	Keterangan
<i>i was standing under a tree when it began to rain</i>	saya berdiri dibawah sebuah pohon ketika telah mulai untuk hujan	Berhasil diterjemahkan
<i>last year at this time, i was attending school</i>	Tahun yang lalu di waktu ini, saya sedang menghadiri sekolah.	Berhasil diterjemahkan
<i>my brother and sister was</i>	saudara(lelaki) saya dan	Terjemahan tidak baik



<i>argueing about something when i walked into the room</i>	saudara(wanita) sedang berdebat tentang sesuatu ketika saya telah berjalan ke ruangan	
<i>i was not hearing the thunder during the storm last night because i was sleeping</i>	saya tidak mendengar petir selama badai malam yang lalu karena saya tidur	Berhasil diterjemahkan
<i>no one could see the thief because he was hiding from the police</i>	tidak ada dapat melihat pencuri karena dia(lelaki) bersembunyi dari polisi	Terjemahan tidak baik

### 5.2.3. Past perfect

Tabel 5.3 Hasil pengujian kalimat *past perfect*

English	Indonesia	Keterangan
<i>my parents had already eaten by the time i got home</i>	orang tuaku telah makan ketika saya pulang	Berhasil diterjemahkan
<i>they had gone before we arrived</i>	mereka telah pergi sebelum kami telah sampai	Berhasil diterjemahkan
<i>i had lived in <b>that</b> house for nine years before i moved to where i live now.</i>	saya telah hidup di yang rumah selama sembilan tahun sebelum saya pindah ke dimana saya hidup sekarang .	Terjemahan tidak baik
<i>john had traveled around the world before he got married last year</i>	john telah bepergian mengelilingi dunia sebelum dia(lelaki)	Berhasil diterjemahkan

	menikah tahun yang lalu	
<i>someone had stolen my book before i went back to the classroom</i>	seseorang telah mencuri buku saya sebelum saya kembali ke ruang kelas	Berhasil diterjemahkan

#### 5.2.4. Past perfect progressive

Tabel 5.4 Hasil pengujian kalimat *past perfect progressive*

English	Indonesia	keterangan
<i>the police had been looking for the criminal for two years before they caught him</i>	polisi telah sedang mencari criminal selama dua tahun sebelum mereka telah menangkap dia(laki-laki)	Berhasil diterjemahkan
<i>her eyes were red because she had been crying</i>	mata nya(wanita) merah karena dia(wanita) telah menangis	Berhasil diterjemahkan
<i>when judy got home, her hair was still wet because she had been swimming</i>	ketika judy pulang, rambut nya(wanita) masih basah karena dia(wanita) telah berenang .	berhasil diterjemahkan
<i>i had been repairing freezer when my friend came to my house</i>	saya telah memperbaiki lemari es ketika teman saya telah datang untuk saya rumah	Berhasil diterjemahkan
<i>wake up! you had been sleeping long enough</i>	bangun! kamu telah tidur cukup lama .	Berhasil diterjemahkan

5.2.5. *Present simple*Tabel 5.5 Hasil pengujian kalimat *present simple*

English	Indonesia	Keterangan
<i>i study for two hours every night</i>	saya belajar selama dua jam setiap malam	Berhasil diterjemahkan
<i>he always eats sandwich for lunch</i>	dia(lelaki) selalu makan sandwich untuk makan siang	Berhasil diterjemahkan
<i>they always arrive on time</i>	mereka selalu sampai tepat waktu	Berhasil diterjemahkan
<i>he needs a pen right now</i>	dia(lelaki) membutuhkan sebuah pena sekarang	Berhasil diterjemahkan
<i>my classes begins at nine</i>	Kelas-kelas saya dimulai pada jam sembilan	Berhasil diterjemahkan

5.2.6. *Present Progressive*Tabel 5.6 Hasil pengujian kalimat *present progressive*

English	Indonesia	Keterangan
<i>john is sleeping right now</i>	john tidur sekarang	Berhasil diterjemahkan
<i>i need an umbrella because it is raining</i>	saya membutuhkan sebuah payung karena hujan	Berhasil diterjemahkan
<i>my father is reading newspapers in the living room and my mom is cooking in the kitchen.</i>	ayah saya membaca koran-koran di ruang tamu dan ibu saya memasak di dapur .	Berhasil diterjemahkan
<i>she is writing another book this year</i>	dia(wanita) menulis buku lain tahun ini	Berhasil diterjemahkan
<i>i am not discussing english with my colleagues at the moment</i>	saya tidak berdiskusi Bahasa Inggris dengan rekan-rekan saya saat ini	Berhasil diterjemahkan



5.2.7. *Present Perfect*Tabel 5.7 Hasil pengujian kalimat *present perfect*

English	Indonesia	Keterangan
<i>they have moved into a new apartment</i>	mereka pindah ke sebuah apartment baru	Berhasil diterjemahkan
<i>we have been here for two weeks</i>	kami telah disini selama dua minggu	Terjemahan tidak baik
<i>she has read the entire book</i>	dia(wanita) membaca seluruh buku	Terjemahan kurang baik
<i>i have never seen the snow</i>	saya tidak pernah telah melihat salju	Berhasil diterjemahkan
<i>they have failed the exam twice</i>	mereka gagal ujian dua kali	Berhasil diterjemahkan

5.2.8. *Present Perfect Progressive*Tabel 5.8 Hasil pengujian kalimat *present perfect progressive*

English	Indonesia	Keterangan
<i>i have been thinking about changing my major</i>	saya telah berpikir tentang mengganti jurusan saya	Berhasil diterjemahkan
<i>i have been sitting here for two hours</i>	saya telah duduk disini selama dua jam	Berhasil diterjemahkan
<i>i have been living here since 1985</i>	saya telah hidup disini sejak 1985	Berhasil diterjemahkan
<i>the little boy is dirty from head to foot because he have been playing in the mud</i>	anak kecil kotor dari kepala ke kaki karena dia(lelaki) telah bermain di lumpur	Berhasil diterjemahkan
<i>she has been waiting for her boyfriend for an hour</i>	dia(wanita) telah menunggu pacar(lelaki)	Berhasil diterjemahkan

<i>already</i>	nya(wanita) selama sebuah jam .	
----------------	------------------------------------	--

### 5.2.9. Future Simple

Tabel 5.9 Hasil pengujian kalimat *future simple*

English	Indonesia	Keterangan
<i>i am going to paint my bedroom tomorrow</i>	saya akan mengecat kamar saya besok	Berhasil diterjemahkan
<i>according to the weather forecast, it will rain tomorrow</i>	menurut ke/untuk cuaca ramalan , akan hujan besok	Terjemahan tidak baik
<i>when bob comes, we will see him</i>	ketika bob datang , kami akan melihat dia(laki-laki)	Berhasil diterjemahkan
<i>after i get home, i will eat dinner</i>	setelah saya pulang , saya akan makan makan malam .	Berhasil diterjemahkan
<i>sue will graduate in june</i>	sue akan lulus di juni	Berhasil diterjemahkan

### 5.2.10. Future Progressive

Tabel 5.10 Hasil pengujian kalimat *future progressive*

English	Indonesia	Keterangan
<i>i will be studying when you come</i>	saya akan belajar ketika kamu datang	Berhasil diterjemahkan
<i>i am going to be studying at the lbery</i>	saya akan belajar di perpustakaan	Berhasil diterjemahkan
<i>tomorrow at this time i will be attending class</i>	besok di waktu ini saya akan menghadiri kelas	Berhasil diterjemahkan
<i>when i get up tomorrow morning, the sun will be</i>	ketika saya bangun besok pagi , matahari akan	Berhasil diterjemahkan

<i>shining</i>	sedang bersinar	
<i>i will be discussing english with my classmates at 2 p.m. next friday</i>	saya akan berdiskusi Bahasa Inggris dengan teman-teman saya pada jam 2 siang jumat selanjutnya .	Berhasil diterjemahkan

5.2.11. Future Perfect

Tabel 5.11 Hasil pengujian kalimat *future perfect*

English	Indonesia	Keterangan
<i>i will have finished my homework by the time i go out on a date tonight</i>	saya akan selesai pekerjaan rumah saya ketika saya keluar di sebuah kencan malam ini	Berhasil diterjemahkan
<i>by the next time i see you, i will have graduated</i>	pada waktu selanjutnya saya melihat kamu , saya akan lulus	Berhasil diterjemahkan
<i>i will have eaten before i go to play badminton tonight</i>	saya akan telah makan sebelum saya pergi untuk bermain bulu tangkis malam ini	Berhasil diterjemahkan
<i>if he keeps getting married, he will have had more than 50 wives before he dies</i>	jika dia(lelaki) tetap menikah , dia(lelaki) akan mempunyai lebih dari 50 istri sebelum dia(lelaki) meninggal	Berhasil diterjemahkan
<i>by the time he gets home, i will have already fallen asleep.</i>	ketika dia(lelaki) pulang , saya akan telah tertidur	Berhasil diterjemahkan



5.2.12. *Future Perfect Progressive*Tabel 5.12 Hasil pengujian kalimat *future Perfect Progressive*

English	Indonesia	Keterangan
<i>i will have been sleeping by the time he gets home</i>	saya akan telah tidur ketika dia(lelaki) pulang	Berhasil diterjemahkan
<i>when professor john retires next month, he will have been teaching for 45 years</i>	ketika professor john pensiun bulan selanjutnya , dia(lelaki) akan telah mengajar selama 45 tahun	Berhasil diterjemahkan
<i>by the year 2000 she will have been living on this earth for 50 years</i>	pada tahun 2000 dia(wanita) akan telah hidup di bumi ini selama 50 tahun	Berhasil diterjemahkan
<i>i will have been swimming for 30 minutes at 9 tomorrow</i>	saya akan telah berenang untuk 30 menit pada jam 9 besok	Berhasil diterjemahkan
<i>my grandmother will have been visiting us tomorrow morning</i>	nenek saya akan telah mengunjungi kami besok pagi	Berhasil diterjemahkan

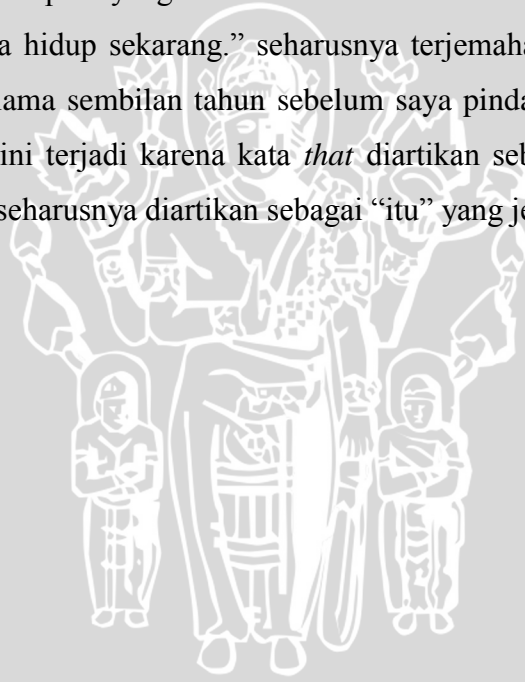
## 5.3 Analisis hasil pengujian

Dari hasil pengujian, dapat diketahui bahwa semua jenis kalimat dapat diterjemahkan. Hal ini berarti fungsi-fungsi yang terdapat dalam aplikasi penerjemah dapat bekerja. Akan tetapi masih terdapat beberapa kalimat yang hasil terjemahannya tidak baik. Contohnya pada kalimat: *my brother and sister was argueing about something when i walked into the room*. Kalimat ini diterjemahkan menjadi: "saudara(lelaki) saya dan saudara(wanita) sedang berdebat tentang sesuatu ketika saya telah berjalan ke ruangan.", seharusnya terjemahannya menjadi: "saudara(lelaki) dan saudara(wanita) saya sedang

berdebat tentang sesuatu ketika saya telah berjalan ke ruangan.” Hal ini terjadi karena pembalikan kata yang dilakukan hanya pada kata *my brother*. Sedangkan yang dibutuhkan dalam kalimat jenis ini adalah meletakkan terjemahan kata *my* setelah kata *brother and sister*.

Kurang baiknya terjemahan juga dapat di temui pada kalimat: *she has read the entire book*. Kalimat ini diterjemahkan menjadi: “dia(wanita) membaca seluruh buku.”, dimana seharusnya terjemahannya: “dia(wanita) telah membaca seluruh buku.” Hal ini disebabkan kata *read* merupakan *irregular verb* yang bentuk masa kini, lampau, dan sempurna nya tidak mengalami perubahan.

Kalimat *i had lived in that house for nine years before i moved to where i live now* juga kurang baik penerjemahannya. Kalimat tersebut diterjemahkan menjadi: “saya telah hidup di yang rumah selama sembilan tahun sebelum saya pindah ke dimana saya hidup sekarang.” seharusnya terjemahannya: “saya telah hidup di rumah itu selama sembilan tahun sebelum saya pindah ke dimana saya hidup sekarang.” Hal ini terjadi karena kata *that* diartikan sebagai “yang” yang jenisnya kata hubung, seharusnya diartikan sebagai “itu” yang jenisnya kata ganti.



## BAB VI

### PENUTUP

#### 5.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, pengujian dan analisis sistem maka dapat diambil kesimpulan sebagai berikut:

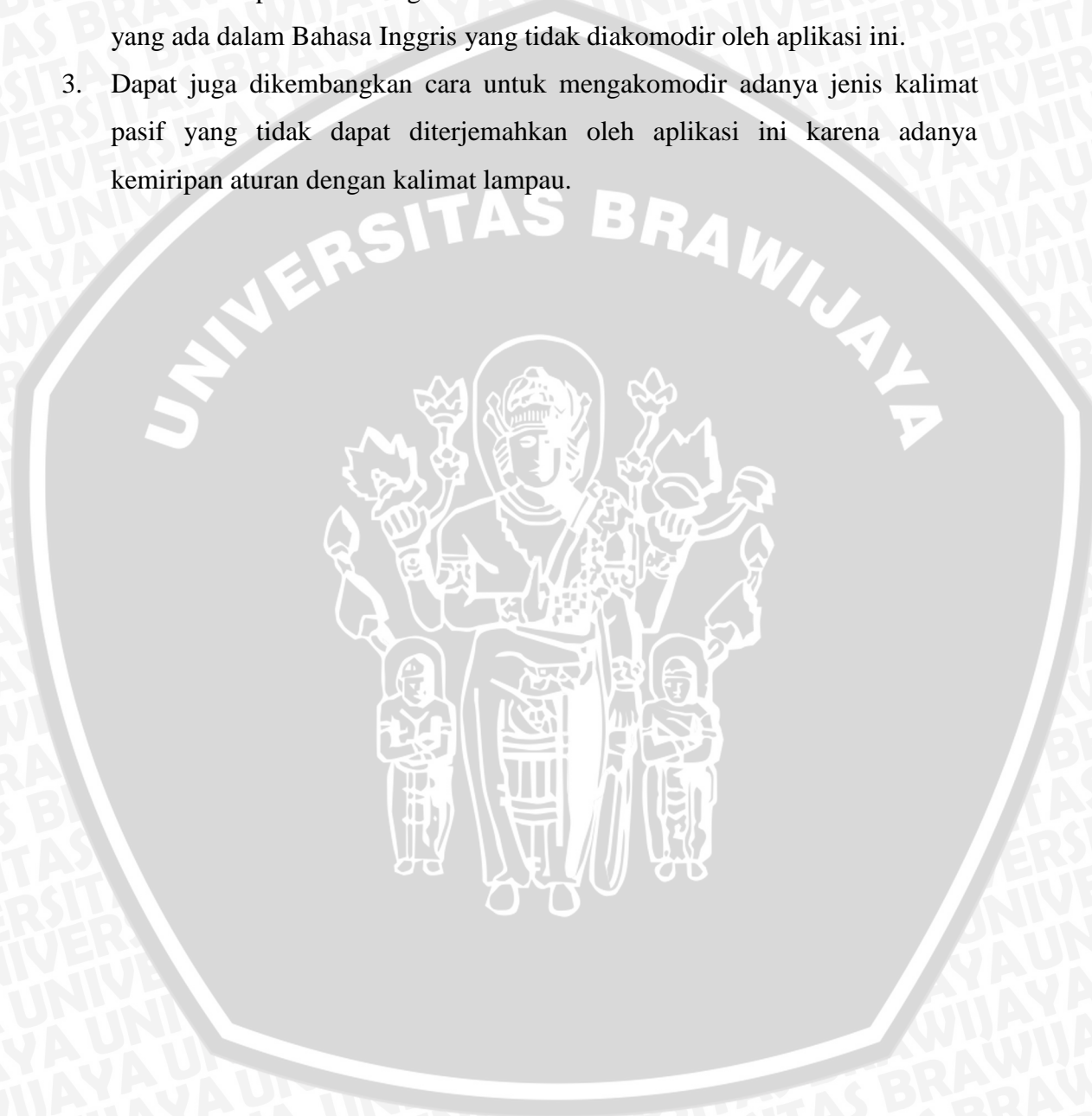
1. Untuk dapat menerjemahkan dengan baik maka *database* yang ada harus mendukung aplikasi penerjemah dan dapat mengakomodir data yang dibutuhkan, oleh karena itu dibuat *database* dengan 4 tabel, yaitu tabel kamus, tabel frasa, tabel disjunct dan tabel kalimat *temporary*.
2. Untuk mengakomodir adanya frasa dalam kalimat maka harus ada fungsi dalam aplikasi untuk menggabungkan kata-kata dan mencocokkan kata-kata tersebut dengan tabel frasa yang sebelumnya telah disediakan.
3. Agar penerjemahan sesuai dengan tata bahasa maka perlu dideteksi jenis kata, imbuhan dan kata yang merupakan ciri dari jenis kalimat
4. Agar fungsi-fungsi yang terdapat dalam aplikasi penerjemah dapat berjalan dengan baik maka perlu fungsi untuk menelusuri kata-kata dalam kalimat *input* mulai awal hingga akhir dan dapat menentukan fungsi yang digunakan pada tiap kata.
5. Proses pengujian menunjukkan bahwa setiap jenis kalimat dapat diterjemahkan, tetapi terdapat kalimat yang hasil terjemahannya kurang baik karena adanya aturan-aturan yang tidak disediakan oleh aplikasi ini.

#### 5.2 Saran

Dalam perancangan dan pembuatan aplikasi penerjemah Bahasa Inggris ke Bahasa Indonesia ini masih terdapat kekurangan dan kelemahan, oleh karena itu masih diperlukan adanya penyempurnaan dalam rangka pengembangan kedepan. Adapun hal yang dapat disempurnakan antara lain:



1. Dapat ditambahkan isi *database* berupa kata Bahasa Inggris dan terjemahannya dan juga frasa yang terdapat dalam Bahasa Inggris agar kalimat yang bisa diterjemahkan lebih banyak.
2. Hasil dari terjemahan aplikasi ini masih ada yang kurang sempurna, oleh karena itu dapat dikembangkan suatu cara untuk menambahkan aturan-aturan yang ada dalam Bahasa Inggris yang tidak diakomodir oleh aplikasi ini.
3. Dapat juga dikembangkan cara untuk mengakomodir adanya jenis kalimat pasif yang tidak dapat diterjemahkan oleh aplikasi ini karena adanya kemiripan aturan dengan kalimat lampau.



## DAFTAR PUSTAKA

- Anonymus, *Idiom*. <http://id.wikipedia.org/wiki/Idiom>. Disakses tanggal 26 mei 2011
- Azar, B.S.1993.*Understanding and Using english Grammar*. Jakarta: Binarupa Aksara & Prentice Hall
- Bimantara, Afif, 2010. *Mengenal Idiom Bahasa Inggris*. <http://aveef.comlu.com/?p=164>. Diakses tanggal 26 mei 2011
- C. J. Date, 1985. *Introduction to Database Systems*. United Kingdom: Addison-Wesley Educational Publishers Inc
- Faiz, Faizal. 2010. *Part of speech*. <http://abudira.wordpress.com/2010/05/13/parts-of-speech>. Diakses tanggal 26 mei2011
- Gordon C. Everest. 1986. *Database Management Objectives, System Functions, & Administration*. Singapore: McGRAW-HILL.
- Indriyawan, Eko. 2008. *Membangun sistem andal dengan delphi*. yogyakarta:ANDI.
- Kao, Anne dan R, Stephen. 2006. *Natural Language Processing and Text mining*. London:Springer
- Korth, Henry F. 1991. *Database System Concepts*. USA: McGraw-Hill International
- Matthew, Neil dan Richard Stones. 2005. *Beginning Database with PostgreSQL : From Novice toProfessional, Second Edition*. New York : Apress.
- Syamsuddin, A.R. 1986. *Sanggar Bahasa Indonesia*. Jakarta: Universitas Terbuka Jakarta.
- Walija. 1996. *Bahasa Indonesia dalam Perbincangan*. Jakarta: IKIP