PEMBANGUNAN WORDNET BAHASA INDONESIA MENGGUNAKAN METODE *AUTOMATIC TRANSLATION*

Dennis Novriandi[1], Ken Kinanti P. S.Kom, M.T[2]
[1,2] Teknik Informatika – Universitas Komputer Indonesia
Jalan Dipatiukur 114-116 Bandung 40132
E-mail : dennissnov@gmail.com[1], guinev_gs@yahoo.co.nz[2]

ABSTRAK

Wordnet adalah jaringan antar makna yang terinspirasi dari memori leksikal manusia atau bisa disebut juga Wordnet adalah suatu wadah untuk membuat sebuah database leksikal online untuk memudahkan Leksikografer atau penggiat bahasa dalam mencari informasi atau hubungan semantik pada suatu lema. Wordnet bisa diterapkan dalam berbagai aplikasi yang membantu manusia mengolah dan mencerna informasi misalnya di bidang *Information Retrieval, Machine Translation,* dan *Natural Language Processing.*

Dalam penelitian ini, pengembangan Wordnet bahasa Indonesia menggunakan *Expand Approach* dengan metode *Automatic translation*. Sumber data masukan berasal dari Princeton Wordnet dan menggunakan MRD dari Cambridge Dictionary. Pada tahap awal dilakukan proses ekstraksi sehingga didapat data dari database PWN yang terstruktur. Tahap kedua dilakukan proses penerjemahan dari bahasa Inggris ke bahasa Indonesia dengan menggunakan MRD dari Cambridge Dictionary.

Pada tahap pertama, hasil ekstraksi Unique Strings sebanyak 100% dari total data awal 155467 dan *Synset* sebanyak 100% dari total data awal sebanyak 117791. Pada tahap kedua pada proses translasi didapatkan Unique Strings sebanyak 15.4% dan *Synset* sebanyak 34.6%. Hal tersebut dikarenakan kualitas dari MRD Sendiri. Banyak lema atau istilah lokal dalam bahasa Inggris yang tidak ada dalam bahasa Indonesia.

Kata kunci : wordnet, wordnet Bahasa Indonesia, Expand approach, Automatic Translation.

PENDAHULUAN

Penelitian Wordnet saat ini banyak diteliti dan dikembangkan diberbagai negara salah satunya Indonesia. Sejak Princeton Wordnet (PWN) [1] pertama kali ditemukan oleh professor psikologi George A. Miller pada tahun 1985, Dari tahun ke tahun banyak pengembangan Wordnet untuk bahasa lain dengan menjadikan PWN sebagai referensi untuk pengembangan. Berbeda dengan pengembangan Wordnet lainya, PWN dibangun secara manual dengan bantuan para Leksikografer, hal itu menjadikan PWN mempunyai akurasi dan kualitas yang tinggi. dikarenakan pembangunan Wordnet dengan cara manual membutuhkan waktu yang panjang dan biaya yang tinggi, maka digunakan metode otomatis dan semi otomatis sebagai salah satu cara untuk mengembangkan Wordnet [2]. Penelitian tentang Wordnet bahasa Indonesia pertama pernah dilakukan di Indonesia pada tahun 2008 di Universitas Indonesia, kendalanya Wordnet bahasa Indonesia belum dikembangkan kembali sampai saat ini. Penelitian tersebut menghasilkan 1203 *Synset* dan 1659 Unique Strings [3]. pada tahun 2010 dilakukan penelitian di Sekolah Tinggi Teknik Indonesia dengan menggunakan metode semi otomatis yang berfokus pada pembuatan *Synset* tanpa mengacu pada Wordnet dari Princeton. Saat ini PWN mempunyai jumlah *Synset* 117659 akan dijadikan resource dan sumber data untuk pengembangan Wordnet bahasa Indonesia. Pengembangan Wordnet bahasa Indonesia belum dilakukan kembali sampai sekarang. Kurangnya sumber daya seperti Leksikografer untuk bahasa Indonesia menjadi masalah bagi para pengembang Wordnet bahasa Indonesia, khususnya untuk mengembangkan *Synset* secara manual.

Untuk perkembangan kamus *online* sendiri saat ini masih bersifat tidak open source dikarenakan keterkaitan dengan pihak ketiga, sehingga keterbatasan sumber data menjadi sulitnya perkembangan kamus khususnya Wordnet bahasa Indonesia. Kegunaan Wordnet bahasa Indonesia bisa dijadikan referensi atau sumber data untuk *Wordsense Disambiguiton, Information Retrieval, Automatic Text Classification, Automatic Text Summarization, Machine Translation* dan lain lain. Namun pada perkembangan Wordnet bahasa Indonesia terkendala dari sumber data yang masih menggunakan data dari luar seperti belum adanya kamus lengkap dari Indonesia untuk proses translasi. Belum adanya resource yang bersifat open source yang bisa digunakan sehingga diperlukan adanya penelitian mengenai pembangunan Wordnet bahasa Indonesia untuk memberikan informasi yang terstruktur mengenai Wordnet.

Berdasarkan pada uraian yang sudah dipaparkan di atas maka dalam penelitian ini akan memberikan kontribusi dengan mengembangkan penelitian sebelumnya dengan cara membangun aplikasi Wordnet bahasa Indonesia dengan menggunakan Expand Approach menggunakan Automtic Translation. Dengan bantuan Kamus dwilingual Inggris-Indonesia untuk translasi lema dari PWN ke

bahasa Indonesia. Penelitian ini diharapkan bisa untuk sumber daya dalam pengembangan bidang teknologi dan menjadi sumber data khususnya khususnya pengembangan Wordnet ke bahasa lain dan untuk bidang NLP.

1. ISI PENELITIAN

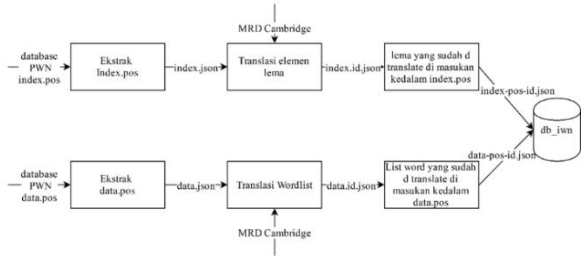
Pengembangan Wordnet menggunakan Expand Approach dengan Atutomatic translation

1.1 Analisis Masalah

Untuk saat ini pengembangan Wordnet bahasa Indonesia belum dilakukan kembali dan belum adanya sumber daya yang sudah bisa digunakan dan bisa dikembangkan dengan sifat open source. Kebutuhan Wordnet untuk informasi masih belum terstruktur dikarenakan pengembangan dari kamus baru dilakukan sehingga dibutuhkan suatu resource untuk mengolah bahasa Indonesia. Pada penelitian sebelumnya Synset yang dihasilkan berjumlah 1203 dan Unique Strings berjumlah 1659.

1.2 Analisis Solusi

Solusi yang diberikan dalam penelitian ini adalah menggunakan Metode Expand Approach dengan Automatic Translation dengan menerjemahkan setiap lema atau kata yang ada pada database Princeton Wordnet (PWN) dengan cara ekstrak database dari Princeton Wordnet dan mengambil setiap relasi semantik yang ada pada PWN. Tahapan pertama yang dilakukan adalah proses pembersihan karakter yang bukan data, seperti license agreement, number version dan copyright notice yang ada pada awal setiap database. Karena format database PWN berupa data dan index dan struktur datanya berbeda untuk itu akan dilakukan dua proses yang berbeda ekstrak database berdasarkan data dan index. Setelah tahap ekstraksi berhasil dilakukan translasi menggunakan Machine Readable Dictionary (MRD) dari Cambridge Dictionary pada setiap lema yang ada pada database PWN. Yang selanjutnya disimpan ke dalam data yang baru. Untuk lebih jelasnya bisa dilihat pada gambar 1 pada Analisis solusi untuk sistem pembangunan Wordnet bahasa Indonesia menggunakan MRD memiliki tahapan ekstrak database PWN, translasi lema, dan konversi format.



Gambar 1. Gambaran Arsitektur Sistem

Gambaran arsitektur sistem di atas dijelaskan sebagai berikut.

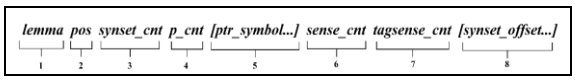
1. Pada tahap awal User memasukan database PWN index.pos yang berformat ASCII ke dalam sistem lalu dilakukan ekstraksi struktur datanya.
2. Selanjutnya dilakukan proses translasi pada elemen pertama yaitu lemma. Translasi menggunakan MRD dari Cambridge Dictionary
3. Lema yang sudah ditranslasi kemudian diurutkan dan disimpan ke dalam ke database.
4. Pada proses ini data masukan adalah database PWN data.pos yang berformat ASCII ke dalam sistem lalu dilakukan ekstraksi dan pengambilan struktur data, hasil keluaran berformat .json
5. Pada tahap ini diambil wordlist pada setiap baris dan dilakukan penerjemahan dari bahasa Inggris ke Indonesia dengan bantuan MRD.
6. Setelah penerjemahan berhasil kemudian mengambil struktur data yang ada pada PWN dan memasukan hasil translasi ke dalam struktur data yang diambil dan disimpan ke dalam database.

1.3 Analisis Data Masukan

Analisis data masukan pada database PWN sendiri mempunyai format khusus dengan mengelompokan data dengan data.pos dan index.pos dimana pos ini adalah kelas kata yaitu *noun,verb,adjective,* dan *adverb*. Format database PWN menggunakan encoding *ASCII* sehingga mudah untuk dibaca oleh manusia dan mesin.

1.3.1 Index.pos

Index file terdiri dari list kata secara alfabet dengan kelas kata masing-masing *database*. Pada setiap baris list kata tersebut mempunyai elemen yang bernama *Synset_offset* yang terhubung dengan *data.pos*. setiap *Synset* mengandung kata. Kata/lema pada *index.pos* mempunyai format *lower case*. Pada *index.pos* setiap *database*nya terdiri *copyright, version number, license agreement* pada awal baris. Berikut adalah struktur data yang ada pada *index.pos.*



Gambar 2 Data Masukan *Index.pos*

berikut penjelasan dari struktur data *index.pos.*

Tabel 1 Tabel Elemen *Index.pos*

No.	Elemen	Penjelasan
1.	*Lemma*	List kata dengan format *lower case ASCII*, spasi digantikan dengan *underscore*(_).
2.	*Pos*	Kelas kata untuk setiap lema, terdiri dari *n(nount), v(Verb), a(adjective), r(adverb).*
3.	*Synset_cnt*	Jumlah dari *Synset* yang terdapat lema/kata di dalamnya. Untuk

		menentukan jumlah *Synset_offset* pada *index.pos*
4.	*P_cnt*	Jumlah dari *Pointer* yang berbeda untuk setiap lema yang terdapat pada *Synset*. Jika tidak ditemukan *Pointer* pada *sysnet* maka *p_cnt* bernilai 0.
5.	*Ptr_symbol*	*List Pointer* yang ada pada *sysnet*. *Pointer* adalah relasi semantic yang ada pada setiap *Synset*.
6.	*Sense_cnt*	Jumlah makna pada setiap *Synset*, jumlah *sense* sama dengan *sysnet_Cnt*
7.	*Tagsense_cnt*	Jumlah makna pada lema yang telah di rank bedasarkan sering munculnya pada relasi semantic.
8.	*Sysnset_offset*	List *offset* pada *data.pos*, setiap *Synset* mengandung lema/kata. Setiap *Synset_offset*

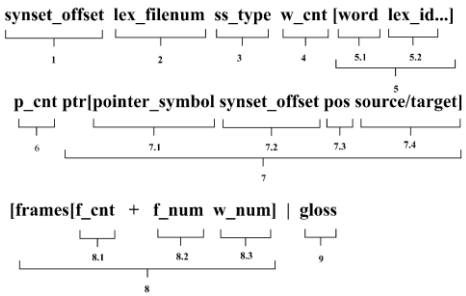
Untuk lebih jelas berikut adalah contoh dari data masukan index.pos.



Gambar 3 Contoh Data Masukan *Index.pos*

1.3.2 Data.pos

File *data.pos* mempunyai informasi yang mengandung anggota *Synset* dari *index.pos*. setiap file *data.pos* diawali dengan *copyright notice, version number* dan *license agreement*. Setiap baris data mempunyai informasi yang sudah disusun oleh Leksikografer untuk *Synset*. Setiap informasi data diawalai oleh 8 *byte offset* atau *address* suatu *Synset.*



Gambar 4 Data Masukan *Data.pos*

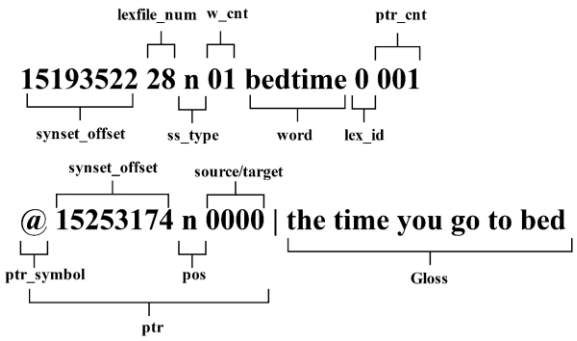
berikut penjelasan dari struktur data *data.pos.*

Tabel 2 Elemen *Data.pos*

No.	Elemen	Penjelasan
1	*Synset_offset*	8 *byte decimal integer* yang berisi alamat untuk informasi *Synset*
2	*Lex_filenum*	2 digit *decimal integer* sesuai dengan data dari Leksikografer yang berisi *Synset*.
3	*Ss_type*	Kelas kata untuk setiap lema, terdiri dari *n(nount), v(Verb), a(adjective), r(adverb).*
4	*w_cnt*	2 digit heksadesimal *integer* yang berisi jumlah kata pada *Synset*.
5.1	*Word*	Kata dengan format *lower case ASCII* . spasi yang diganti dengan *underscore*(_). Khusus pada *data.adj* kata diikuti dengan *syntactic* marker jika datanya secara khusus telah didefinisikan di data Leksikografer.
5.2	*Lex_id*	Satu digit *hexadecimal integer* yang ada pada kata. Mengacu pada file leksikographer. Biasanya *lex_id* dimulai dari angka 0. 0 terasuk status *default* dan tidak mempunyai data Leksikografer.
6	*P_cnt*	Tiga digit *decimal integer* yang berisi tentang jumlah *Pointer* pada suatu *Synset*. *P_cnt* bernilai 000 maka *Synset* tidak mempunyai *Pointer*.
7	*Ptr*	*Pointer* pada suatu *Synset* , mempunyai struktur : *Pointer*_symbol *offset pos* source/target
7.1	*Ptr_symbol*	List *Pointer* yang ada pada *sysnet*. *Pointer* adalah relasi semantic yang ada pada setiap *Synset*.
7.2	*Synset_offset*	8 byte *decimal integer* yang berisi alamat untuk informasi *Synset*
7.3	*Pos*	Kelas kata untuk setiap lema, terdiri dari *n(nount), v(Verb),*

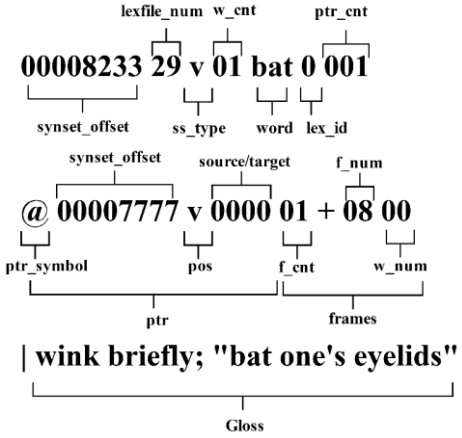
		a(adjective), r(adverb).
7.4	*Source/Target*	*Source*/*Target* adalah leksikal dan semantic *Pointer*. Berjumlah 4 digit *character* , 2 digit pertama adalah heksadesimal *integer* 2 digit pertama mengindikasikan jumlah kata pada *Synset*. Dan 2 digit terakhir mengindikasi jumlah kata pada target *Synset*.
8	*Frames*	*Frames* hanya ada pada *data.verb*, mempunyai bentuk *F_cnt + f_num w_num*
8.1	*F_cnt*	Dua digit *decimal integer* yang menginidikasi jumlah *frames* yang ada pada suatu baris.
8.2	*F_num*	Dua digit heksadesimal *integer* yang menginidikasi nomor suatu *frames*
8.3	*W_num*	Dua digit heksadesimal *integer* yang menginidikasi jumlah kata yang mengandung *frames* ,
9	Gloss	Setiap *Synset* mempunyai *gloss* atau definisi , dipisahkan oleh karakter *pipe* "|". Pada elemen *gloss* terdapat definisi dan contoh kalimat

Berikut adalah contoh data masukan *data.pos.*



Gambar 5 Contoh Data Masukan *Data.pos*

Ada sedikit perbedaan untuk *data.verb* yaitu adanya elemen *frames*, Berikut adalah Contoh *data.verb* untuk *frames*.



Gambar 6 Contoh Data Masukan *Data.verb*

1.4 Analisis Proses

Analisis proses yang dilakukan adalah ekstraksi dan translasi sebagai proses utama dan konversi data. Karena struktur data pada *data.pos* dan *index.pos* berbeda maka dilakukan proses ekstrak yang berbeda untuk itu dilakukan 2 proses yang berbeda.

1.4.1 Ekstraksi Index.pos

Tahap ini berfokus pada pengambilan struktur data pada index.pos untuk mendapatkan struktur yang terdiri dari lemma, pos, Synset_cnt, p_cnt [ptr_symbol...] sense_cnt, tagsense_cnt dan Synset_offset [Synset_offset...]. data masukan berasal dari Princeton Wordnet (PWN) yang berformat ASCII. PWN sendiri membagi database nya menjadi empat bagian yang masing-masing dikelompokan berdasarkan kelas kata yaitu index.noun, index.verb, index.adverb, dan index.adjective Contoh data masukan yang diambil dari PWN adalah sebagai berikut

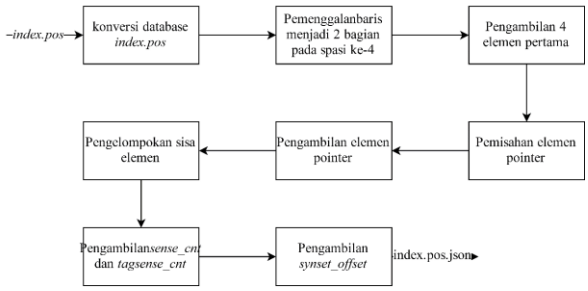
Tabel 3 Data *Index.pos*

```
commerce n 3 6 @ ~ %p = + - 3 1
01092370 08146250 07152330

commando n 2 3 @ ~ ; 2 1 09961754
08421330
```

Tabel di atas dari *database* PWN yang menjadi sumber data yang akan dilakukan ekstrak dan pengambilan struktur data. Tujuanya sistem yang dibuat akan mengambil setiap bagian struktur dan memasukan ke dalam tabel di *database*

Berikut adalah tahapan yang dilakukan dalam ekstraksi database *index.pos*



Gambar 7 Skema Ekstraksi Database *Index.pos* dari PWN

1.5 Ekstraksi *Data.pos*

Setelah pengambilan *database index.pos* tahap selanjutnya adalah pengambilan *data.pos* yang merupakan berisi data dari *index*. Tahap ini berfokus pada pengambilan struktur data pada *data.pos* untuk mendapatkan struktur dari *Synset_offset, lex_filenum, ss_type, w_cnt, word, lex_id, [word lex_id...], p_cnt, [ptr...], [frames...]* , | *gloss* , . data masukan berasal dari *Princeton Wordnet* (PWN) yang berformat *ASCII*, sama seperti *index* PWN membagi *database* nya berdasarkan *pos* atau kelas kata yaitu *data.noun, data.verb, data.adv*, dan *data.adj*. contoh data masukan yang diambil dari PWN adalah sebagai berikut.

Tabel 4 Contoh Data Masukan *Data.pos*

```
08497858 14 n 01 blue 0 002 @
08024893 n 0000 ~ 08411492 n 0000
| any organization or party whose
uniforms or badges are blue; "the
Union army was a vast blue"

08497673 14 n 01 vote 0 000 | a
body of voters who have the same
interests; "he failed to get the
Black vote"
```

Data masukan pada tabel di atas adalah *data.pos* dari *database* PWN yang menjadi sumber data yang akan dilakukan ekstrak dan pengambilan struktur data. Tujuanya sama seperti *index* untuk mengambil setiap elemen. Untuk struktur datanya lebih kompleks dan lebih sulit dari *index.pos* untuk setiap elemenya tidak semua sama ada setiap baris. beberapa perbedaan pada setiap data seperti.

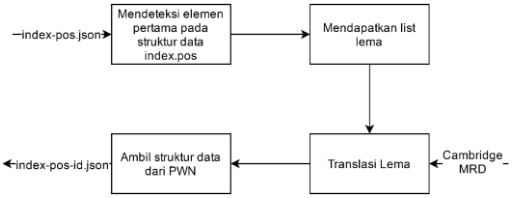
1. *[frames...]*, yang posisinya ada sebelum *gloss* hanya ada pada *data.verb*, dan tidak ada pada data yang lain.
7. Untuk beberapa kasus ada baris yang tidak mempunyai definisi kata
8. Beberapa baris tidak ada contoh kalimat.
9. Beberapa baris tidak mempunyai *Pointer*.

1.6 Translasi Index.pos dan Data.pos

Setelah proses ekstrasi maka dilakukan penerjemahan setiap *Synset* pada *database* yang sudah diekstrak. Data masukan berupa hasil ekstrak *index.pos* dan *data.pos* dengan format *.json*. Proses translasi dengan diterjemahkan menggunakan *Machine readable Dictionary* (*MRD*) dari *Cambridge Dictionary*. Alasan menggunakan *MRD Cambridge Dictionary* dikarenakan jumlah datanya yang bisa diambil dari *website*. Pada proses ini sistem akan mengecek apakah lema yang akan diterjemahkan tersebut ada terjemahanya atau tidak, jika didapatkan terjemahanya maka elemen pertama tersebut akan di update dan diisi dengan hasil terjemahan bahasa Indonesia, sedangkan jika tidak ditemukan terjemahannya maka lema tersebut diisi dengan isian kosong. Hal tersebut berguna untuk validasi seberapa banyak jumlah lema yang telah berhasil diterjemahkan.

1.6.1 Translasi Index.pos

Pada proses translasi *index.pos* elemen yang diambil adalah elemen pertama dari struktur data yaitu elemen lema. Elemen lema merupakan *Unique String* pada *database Princeton Wordnet* yang digunakan untuk pencarian dalam anggota *Synset*. Proses Translasi yang dilakukan adalah sebagai berikut.

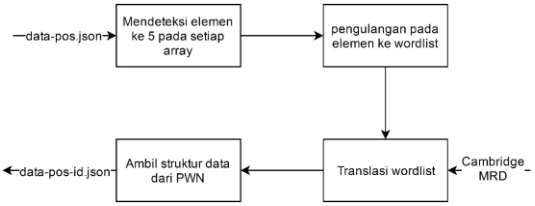


Gambar 8 Blok Diagram Proses Translasi *Index.pos*

Seperti kita liat struktur data yang diambil adalah elemen pertama yaitu lema, pada contoh di bawah terdapat list *unique string* yang diambil yaitu *commerce* dan *commando*. Sistem melakukan translasi lalu kemudian dihasilkan hasil translasi untuk target bahasa dan dimasukan kembali ke dalam struktur data dari *Princeton Wordnet*.

1.6.2 Translasi Data.pos

Pada Proses translasi *data.pos* diambil dari hasil ekstraksi *data.pos* yaitu elemen ke-5 yaitu *Wordlist*. Untuk list kata Karena jumlah kata dalam setiap baris bisa lebih dari satu maka dilakukan pengulangan untuk melakukan translasi setelah itu proses pengambilan struktur data dari hasil ekstraksi. Untuk lebih jelasnya bisa dilihat pada diagram blok di bawah ini.



Gambar 9 Blok Diagram Translasi *Data.pos*

Pada elemen ke-5 pada setiap *array* dilakukan pengulangan jika kata mempunyai lebih dari satu. Lalu dilakukan translasi dari *MRD Cambridge Dictionary* pada list kata tersebut, Kemudian list kata

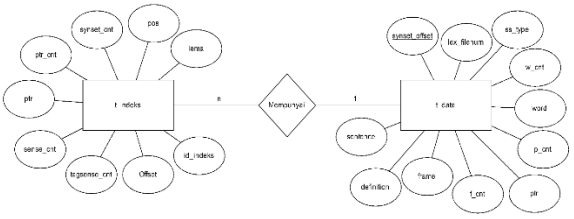
tersebut dimasukan ke dalam struktur data masing-masing untuk setiap baris.

1.7 Analisis Basis Data

Analisa ini akan digambarkan dengan ERD dan dijelaskan dalam kamus data ERD.

1.7.1 Entity Relationship Diagram

Sistem yang akan dibangun akan menggunakan skema *ERD* di bawah ini.



Gambar 10 *Entity Relationship Diagram* (*ERD*)

1.7.2 Kamus Data *ERD*

Penjelasan pada gambar E*ntity relationship Diagram* di atas menggunakan kamus data *ERD*, untuk lebih jelas bisa dilihat di bawah ini.

Tabel 5 Kamus Data *ERD*

Entitas	Atribut
t_indeks	*id_indeks, lema,pos,Synset_cnt,ptr_cnt, ptr_sense_cnt, tagsense_cnt, offset.*
t_data	*Synset_offset, lex_filenum, ss_type, w_cnt, word, p_cnt, ptr, f_cnt, frames , definition, sentence.*

1.8 Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan non-fungsional dilakukan untuk mengetahui kebutuhan spesifikasi untuk sistem yang meliputi *Hardware* dan *software*, Analisa ini dibutuhkan untuk mempermudah menganalisis sebuah sistem.

1.8.1 Analisis Kebutuhan Perangkat Keras Pembangunan Sistem

Kebutuhan perangkat keras (*hardware)* untuk pembangunan ini adalah spesifikasi minimum untuk menjalankan sistem ini, kendala yang akan dihadapi adalah proses akan memakan waktu lama. Untuk spesifikasi minimum perangkat keras adalah sebagai berikut.

Tabel 6 Spesifikasi Minimum Perangkat Keras

Perangkat Keras	Kebutuhan Minimum
Prosesor	Intel Pentium IV 3,0 GHz atau AMD Athlon X4
RAM	1Gb

1.8.2 Analisis Kebutuhan Perangkat Lunak Pembangunan Sistem

Analisis kebutuhan perangkat lunak (*Software*) untuk pembangunan sistem ini adalah sebagai berikut.

Tabel 7 Spesifikasi Minimum Perangkat Lunak

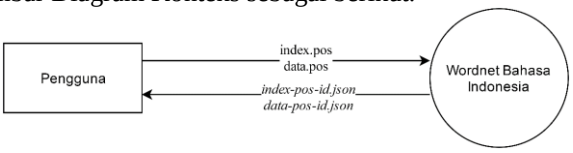
Perangkat Lunak	Versi Minimum Perangkat Lunak
Sistem Operasi	Windows XP
Web Browser	Google Chrome,mozilla,opera,safari

1.9 Analisis Kebutuhan Fungsional

Tujuan dari analisis ini adalah untuk menganalisis proses yang akan diterapkan dalam sistem yang akan dibangun menggunakan Diagram Konteks, Pembuatan *DFD*, dan Spesifikasi Proses.

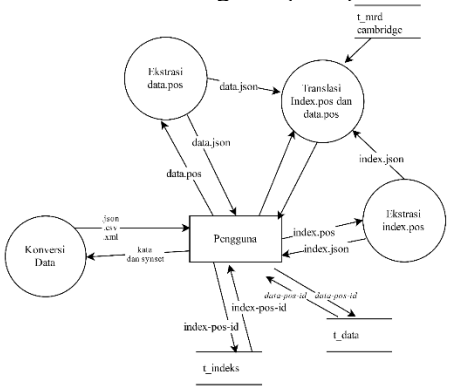
1.9.1 Diagram Konteks

Diagram konteks yang terdapat dalam Wordnet bahas Indonesia yang dibangun dapat dilihat pada gambar Diagram Konteks sebagai berikut.



Gambar 11 Diagram Konteks

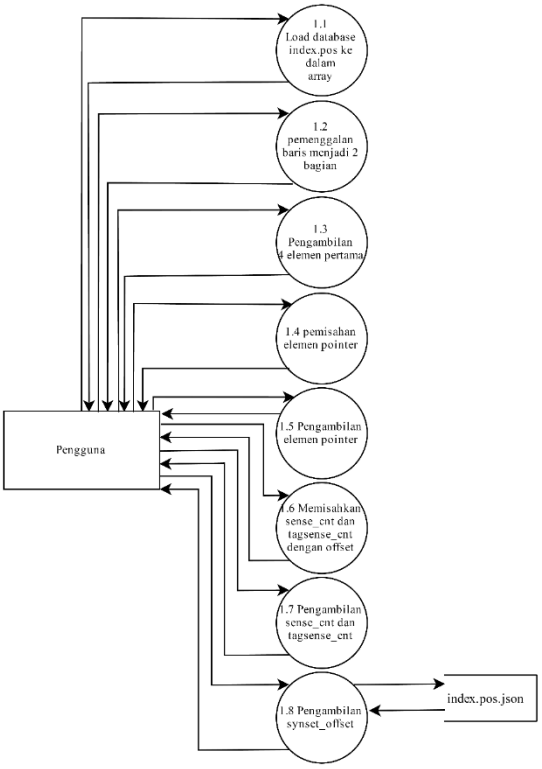
1.9.2 Data Flow Diagram (DFD) Level 1



Gambar 12 Data Flow Diagram (DFD) Level 1

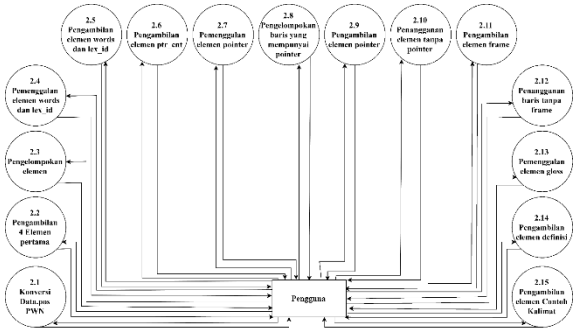
1.9.3 Data Flow Diagram (DFD) Level 2 Index.pos dan Data.pos

Data yang sudah masuk ke dalam pengambilan *index.pos* dan *data.pos* yang digambarkan dalam bentuk *DFD* merupakan alur dari data masuk sampai menjadi data yang dihasilkan. Mulai dari pengekstrakan *database* dari *index.pos* dan *data.pos*. Hasil dari pengambilan *database* ini disimpan berupa *.json* yang akan diproses untuk tahap selanjutnya berupa translasi setiap lema. Data masukan mempunyai ekstensi sendiri berupa *pos* atau kelas kata yang berformat *ASCII*, data yang akan dihasilkan berupa *.json*. berikut adalah *DFD* dari pengambilan *database index.pos* dan *data.pos*.



Gambar 13 DFD Level 2 Proses Ekstraksi *Index.pos*

Berikut adalah DFD Level 2 proses ekstraksi untuk data.pos



Gambar 14 DFD Level 2 Ekstraksi Data.pos

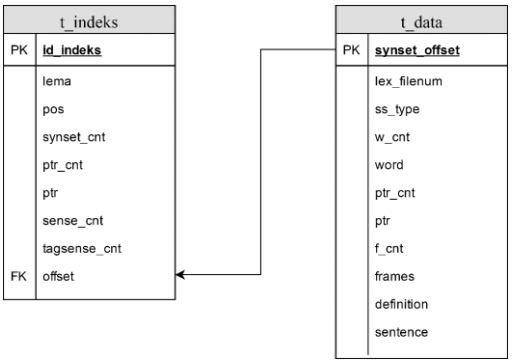
1.9.4 Data Flow Diagram (DFD) Level 2 Translasi Index.pos dan Data.pos

1.10 Perancangan Basis Data

Perancangan basis data menentukan isi dan pengaturan data yang dibutuhkan untuk mendukung berbagai rancangan sistem, tujuanya untuk memenuhi informasi yang berisikan kebutuhan dalam pembangunan Wordnet bahasa Indonesia.

1.10.1 Skema Relasi

Skema relasi memberikan hubungan antar tabel secara lengkap, termasuk di dalamnya terdapat atribut. Hal ini diperlukan untuk menyimpan informasi mengenai objek-objek yang direpresentasikan dalam sebuah basis data. Tabel yang dibangun untuk Wordnet bahasa Indonesia mengambil struktur data dari *database Princeton Wordnet* (PWN). Sehingga terdapat dua tabel yaitu tabel indeks untuk list *Synset* dan tabel data untuk data dari *Synset* tersebut. Skema relasi yang akan dibangun ditunjukan pada gambar berikut.



Gambar 15 Skema Relasi

1.10.2 Struktur Tabel

Tabel-tabel yang terdapat dalam perancangan basis data pada sistem yang akan dibangun dijelaskan seperti berikut.

Tabel 8 Struktur Tabel t_indeks

Nama Field	Tipe Data	Ukuran	Kunci	Keterangan
id_indeks	*Integer*	11	PK	*NOT NULL*
lemma	*Varchar*	250		*NOT NULL*
Pos	*Enum*	(‘a’,’n’, ’v’,’r’)		*NOT_NULL*
Synset_cnt	*Integer*	3		*NOT NULL*
Ptr_cnt	*Integer*	3		*NOT_NULL*
Ptr	*Varchar*	250		
Sense_cnt	*Integer*	3		*NOT_NULL*
Tagsense_cnt	*Integer*	3		*NOT_NULL*
offset	*Varchar*	250	FK	

Tabel 9 Struktur Tabel t_data

Nama Field	Tipe Data	Ukuran	Kunci	Keterangan
Synset_offset	*Varchar*	8	PK	*NOT NULL*
Lex_filenum	*Int*	3		*NOT NULL*
Ss_type	*Enum*	(‘a’,’n’,’v’,’s’,’r’)		*NOT_NULL*
w_cnt	*Int*	3		*NOT_NULL*
Word	*Text*			
P_cnt	*Int*	3		*NOT_NULL*

Ptr	*Text*			
F_cnt	*Int*	3		*NOT_NULL*
Frames	*Text*			
Definition	*Text*			
Sentence	*Text*			

1.11 Pengujian Akurasi

Pada pengujian akurasi akan dihitung nilai dari jumlah Synset dari hasil Unique Strings yang didapat. Jumlah Synset yang ada pada Princeton Wordnet adalah 117659 dari semua jenis kelas kata. Data yang diambil dari index.noun, index.verb, index.adj, index.adv. Berikut adalah tabel hasil ekstraksi dan hasil translasi dari pengujian yang didapat.

Tabel 10 Hasil Pengujian

Kategori	**Data Awal**		**Hasil Ekstraksi**		**Hasil Translasi**	
	Unique Strings	***Synset***	***Unique Strings***	***Synset***	***Unique Strings***	***Synset***
Noun	117953	82192	117953	82192	11944	20992
Verb	11540	13789	11540	13789	5283	10290
Adjective	21499	18185	21499	18185	4755	7366
Adverb	4475	3625	4475	3625	1982	2126
TOTAL	**155467**	**117791**	**155467**	**117791**	**23964**	**40774**

Pada pengujian akurasi akan dihitung nilai dari jumlah *Synset* dari *database Princeton Wordnet*. Jumlah *Synset* yang ada pada *Princeton Wordnet* adalah 117659 dari semua jenis kelas kata dan jumlah *Unique Strings* total 115467.Dalam pengujianya jumlah *Synset* yang berhasil didapatkan dengan menggunakan pendekatan Expand Approach dengan metode otomatis adalah 40774 *Synset* dan 23964 *Unique Strings*

Pada tahap pertama yaitu ekstraksi *database* dari PWN didapat hasil akurasi 100 %, semua data beserta strukturnya berhasil didapatkan. Pada Tahap kedua dilakukan translasi untuk mendapatkan *Synset* dan *Unique Strings* ke dalam target bahasa, Untuk memperoleh presentase pembangunan sysnset dari hasil translasi oleh *MRD Cambridge Dictionary*. Maka perhitungan yang dilakukan membagi hasil *Synset* yang berhasil di translasi dengan jumlah *Synset*s dalam PWN yang selanjutnya dikalikan dengan 100 %. Untuk detail nya adalah sebagai berikut.

Jumlah *Unique Strings Princeton Wordnet* (US-PWN): 155467

Jumlah *Unique Strings* yang dihasilkan sistem (US-IWN) : 23964

Akurasi *Unique Strings*

$$= \frac{\text{US-IWN}}{\text{US-PWN}} \times 100\%$$

$$= \frac{23964}{155467} \times 100\%$$

$$= 15.4\ \%$$

Penelitian yang dilakukan menghasilkan nilai akurasi sebesar 15.4 %

Sedangakan untuk mendapatkan akurasi *Synset* adalah sebagai berikut.

Jumlah *Synset Princeton Wordnet* (S-PWN): 117791

Jumlah *Synset* yang dihasilkan sistem (S-IWN) : 40774

Akurasi *Synset*

$$= \frac{\text{S-IWN}}{\text{S-PWN}} \times 100\%$$

$$= \frac{40774}{117791} \times 100\%$$

$$= 34.6\ \%$$

Penelitian yang dilakukan menghasilkan nilai akurasi sebesar 34.6 %.

1.11.1 Analisis Hasil Pengujian

Berikut adalah analisis hasil pengujian akurasi yang dilakukan dalam pengembangan Wordnet bahasa Indonesia.

Data yang dihasilkan adalah dari kategori noun, verb, adj, dan adverb hasil yang didapat untuk Synset adalah sebesar 40774 dari total 117791 dan Unique Strings dihasilkan 23964 dari 155467. Pada proses translasi bisa dilihat pada kategori noun yang berdampak besar untuk presentase hasil uji hanya didapat sekitar 10% untuk Unique Strings hal tersebut dikarenakan tidak ada nya kata terjemahan dari hasil translasi menggunakan MRD Cambridge Dictionary dikarenakan bentuk kata dari noun kebanyakan adalah istilah kata benda yang bersifat regional dalam Bahasa Inggris. Banyak istilah-istilah Medis, Kimia, dan Istilah untuk kamus khusus lainya yang tidak dapat diterjemahkan dan kata majemuk dalam Bahasa inggris sangat sedikit terjemahan Bahasa Indonesia nya.

2. PENUTUP

Pada Bagian ini menjelaskan tentang kesimpulan dan saran dari penelitian yang dilakukan.

2.1 Kesimpulan

Sistem yang dibangun dalam hal ini adalah pengembangan Wordnet bahasa Indonesia telah berhasil dibuat. Dan telah tersedianya suatu database leksikal secara *open source*. Pada tahap pertama mendapatkan hasil ekstraksi *Unique Strings* sebanyak

155467 dari total data awal 155467 dan *Synset* sebanyak 117791 dari total data awal sebanyak 117791. Pada tahap kedua pada proses translasi didapatkan *Unique Strings* sebanyak 23964 dan *Synset* sebanyak 40774. Bisa dilihat pada hasil ekstraksi didapatkan hasil ekstraksi sepenuhnya pada tahap pertama, dan pada tahap kedua terjadi penyusutan hasil dikarenakan kualitas dari *MRD*. Banyak lema seperti kata "batman" yang tidak ada terjemahanya dan banyak istilah-istilah lokal untuk bahasa Inggris yang tidak ada di bahasa Indonesia.

2.2 Saran

Saran untuk penelitian berikutnya adalah menambahkan elemen makna dari KBBI dan penambahan peningkatan *Unique Strings* dan *Synset* dengan metode otomatis atau pembangunan *Synset* secara semi otomatis menggunakan thesaurus dan KBBI.

DAFTAR PUSTAKA

[1] G. A. M. (. W. A. L. D. f. English., "WordNet: A Lexical Database for English.," WordNet: A Lexical Database for English. Communications of the ACM Vol. 38, No. 11: 39-41. , 1995.

[2] P. Vossen, EuroWordNet: A Multilingual Database with Lexical Semantic, Publishers, 1998, 179 pp; hardbound,, 1998.

[3] A. A. a. R. M. Desmond Darma Putra, Building indonesian Wordnet, Universitas Indonesia, 2008.

[4] B. M. E. P. Drs. Sumardi Suryabrata, Metodelogi penelitian.

[5] I. Sommerville, Software Engineering (Rekayasa Perangkat), Erlangga, 2011.

[6] D. A. Chaer, Tata Bahasa Praktis Bahasa Indonesia, Jakarta: Rineka Cipta, 2011.

[7] M. Dr. R Kunjana Rahardi, Bahasa Indonesia Perguruan Tinggi, Erlangga, 2015.

[8] R. beckwith, G. A. Miller and R. tengi, Design and Implementation of the WordNet Lexical Database, 1993.

[9] R. A dan M. Salahudin, dalam Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek, Bandung, Informatika Bandung, 2014.

[10] c. felabaun, English Verb as semantic net, 1993.

[11] A. Solichin, Pemrograman Web dengan PHP dan MySQL, 2016.

[12] M. N. S. S. B. Nurril Hirfana, Creating the Open Wordnet Bahasa, Nanyang university.