EKSTRAKSI INFORMASI DAN SQL PARSER UNTUK QUERY SQL DARI TEKS BERBAHASA INDONESIA

Nendi Isharmawan[1], Galih Hermawan, S.Kom., M.T.[2]

[1]Universitas Komputer Indonesia
Jl. Dipatiukur No. 112-116, Bandung 40132
E-mail : nisharmawan@gmail.com[1], galih.hermawan@gmail.com[2]

ABSTRAK

Informasi adalah bagian penting dalam kehidupan sehari-hari, dimana informasi tersebut diharapkan bisa bermanfaat dan dapat diperoleh dengan mudah dan lebih cepat. Salah satu media penyimpanan informasi yaitu *database* yang ada pada komputer salah satunya adalah *Database Management System*. Pengolahan struktur dan data yang ada di DBMS menggunakan *Structured Query Language* (SQL) yang dibagi dua yaitu DDL dan DML. Pengaksesan *database* dan tabel serta data nya memiliki format baku atau disebut *syntax*. Namun tidak semua data masukan yang didapatkan untuk mencari informasi dalam bentuk suatu *syntax* SQL. Seringkali data masukan berupa bahasa alami manusia. Bahasa alami manusia tentunya memiliki format yang berbeda dengan *syntax* SQL.

Natural Language Processing merupakan salah satu bidang ilmu kecerdasan buatan yang mempelajari komunikasi antara masusia dengan komputer melalui bahasa alami. Aplikasi *Natural Language Processing* (NLP) yang dibangun untuk menterjemahkan bahasa alami manusia kedalam bahasa *sql* yang dapat langsung dieksekusi oleh sistem agar mendapatkan informasi dari sumber yang pasti. Berdasarkan pengujian *Blackbox*, aplikasi ini dapat mempermudah suatu media pencari informasi dalam mencari informasi dengan persentasi keberhasilan sebersar 90,47% aplikasi ini dapat mengelola beberapa jenis *query*, yaitu *query* tanpa kondisi, *query* dengan kondisi, *query* dua kondisi, *query* banyak tabel tanpa kondisi, *query* banyak tabel dengan kondisi, *query* agregasi.

Kata Kunci : Informasi, *Query*, NLP, Bahasa alami.

1. PENDAHULUAN

Informasi adalah bagian penting dalam kehidupan sehari-hari, dimana informasi tersebut diharapkan bisa bermanfaat dan dapat diperoleh dengan mudah dan lebih cepat. Salah satu media penyimpanan informasi yaitu database yang ada pada DBMS komputer. DBMS merupakan engine atau software yang digunakan untuk mengelola database. Pengolahan struktur dan data yang ada di DBMS menggunakan Structured Query Language (SQL) yang dibagi dua yaitu DDL (*Data Definition Language*) dan DML (*Data Manipulation Language*).

DDL digunakan untuk mengolah struktur database yang meliputi pembuatan database, pembuatan tabel, dan pengubahan struktur tabel. Sedangkan DML digunakan untuk mengolah data yang ada ditabel meliputi penambahan data, pengubahan data, penghapusan data, dan pemilihan atau preview data. Pengaksesan database dan tabel serta data nya memiliki format baku atau disebut syntax. Namun tidak semua data masukan yang didapatkan untuk mencari informasi dalam bentuk suatu syntax SQL. Seringkali data masukan berupa bahasa alami manusia. Bahasa alami manusia tentunya memiliki format yang berbeda dengan syntax SQL.

Penelitian ini fokus terhadap *natural language processing* (pengolahan bahasa alami) sebagai jembatan untuk mengakses data yang ada di DBMS MySQL. Agar penelitian ini menjadi lebih terarah beberapa batasan masalah ditetapkan sebagai berikut:

1. *Structured Query Language* yang diteliti pada DBMS MySQL yaitu khusus *query* SELECT
2. *Query* SELECT yang dilakukan memiliki 6 tipe, yaitu, *query* tanpa kondisi, *query* dengan kondisi, *query* dua kondisi, *query* banyak tabel tanpa kondisi, *query* banyak tabel dengan kondisi, *query* agregasi.

Tujuan yang ingin dicapai dari penelitian ini adalah :

1. Menterjemahkan bahasa alami menjadi *syntax* SQL.
2. Mengetahui presentasi akurasi dalam penerapan *Natural Language Processing* dalam penterjemah bahasa alami kedalam *syntax* SQL.

2. ISI PENELITIAN

Pada tahap ini akan membahas mengenai masalah yang akan dianalisis. Beberapa hal yang terdapat dalam tahap ini yaitu mengenai tinjauan pustaka, pembahasan dan hasil dari penelitian

2.1 Tinjauan Pustaka

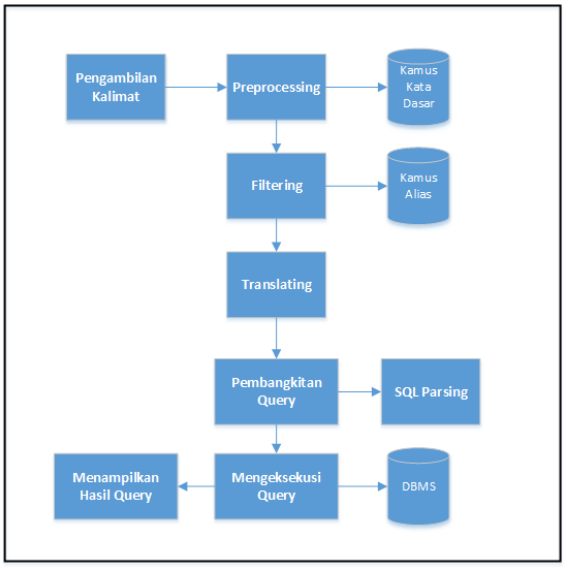
Beberapa penelitian mengenai alat bantu untuk media pencari informasi adalah membuat penterjemah bahasa alami, karena perbedaan antara bahasa alami manusia dan bahasa yang digunakan untuk mengakses database. Seperti penelitian Sri

Hartati dan Eri Zuliarso, melakukan penelitian untuk pengolahan bahasa alami yang diterapkan pada basisdata XML, dimana penelitian ini dilakukan pada database sistem informasi pengolahan data buku [4]. Setyawan Wibisono, melakukan hal serupa dengan melakukan penelitian bahasa alami untuk basisdata akademik dengan format data XML dengan menambahkan pola-pola kalimat yang dapat menjadi referensi pengguna aplikasi tersebut [2]. Sariyun Naja Anwar melakukan penelitian dalam pembuatan query pada basisdata soal ujian yang dilakukan di Universitas STIKUBANK, seperti penelitian sebelumnya, pada penelitian ini juga menggunakan basisdata XML dalam media penyimpanan informasi [5]. Muslim Aries melakukan penelitian penterjemah bahasa alami dengan mesin virtual atau yang sering dikenal sebagai *chatbot* dimana fungsinya hampir sama yaitu untuk mendapatkan informasi yang telah diberikan sebelumnya kepada mesin virtual tersebut [1]. Herry Sujani melakukan penetilitan mengenai bahasa alami menjadi pola sintaks sql yang hanya memiliki dua pola yaitu pola sintaks pemanggilan dengan atribut terpilih dan tidak ada atribut terpilih keduanya tanpa kondisi [3].

Dari penelitian yang ada, penelitian yang akan dilakukan ini fokus kepada pembuatan suatu mesin penterjemah yang dapat digunakan pada setiap *database* pada DBMS MySQL.

2.2 Metode Pengolahan Bahasa Alami

Untuk melakukan penterjemahan bahasa alami menjadi suatu *query* perlu ada beberapa tahapan yang harus dilakukan dapat dilihat pada Gambar 2.1.

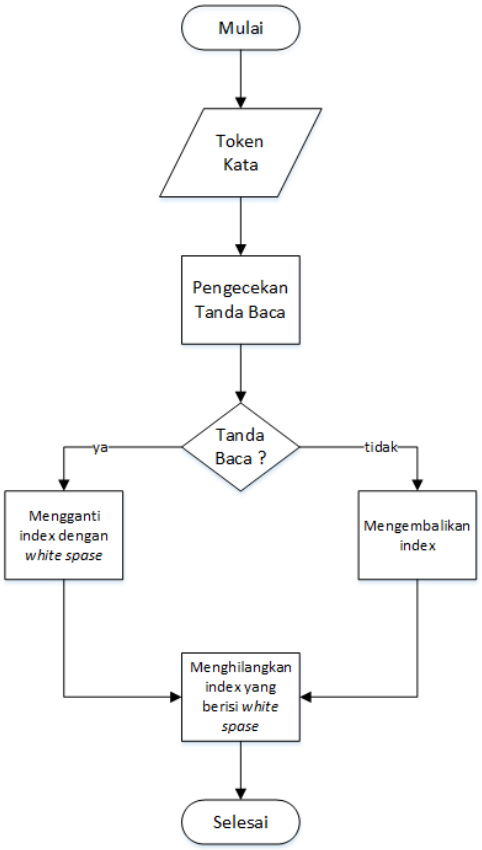


Gambar 2. 1 Tahapan Pengolahan Bahasa alami

1. Kalimat masukan
   Mendapatkan bahasa alami berupa kalimat yang dimasukan oleh pengguna yaitu :
   "Perlihatkan nama mahasiswa"

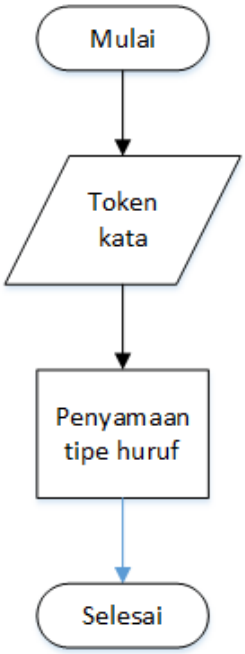
2. *Scanning*
   Pada tahap ini dilakukan tahap *scanning* yaitu memindai kalimat bahasa alami yang dimasukan lalu melakukan pemenggalan kalimat menjadi kata perkata. Setelah dipenggal menjadi :
   "Perlihatkan","nama","mahasiswa"

3. *Cleansing*
   Hasil dari proses diatas didapatkan daftar token yang berbentuk *array*, akan dilakukan proses menghilangakan tanda baca yang tidak digunakan seperti ". ! , ?". Proses dapat dilihat pada Gambar 2.2.



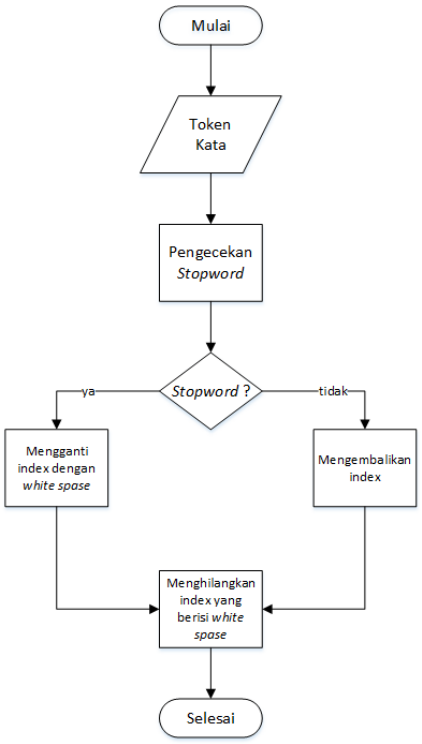
Gambar 2. 2 Tahapan *Cleansing*

4. *Case Folding*
   Hasil dari proses sebelumnya akan dilakukan proses menyamakan *case* menjadi huruf kecil semua untuk setiap tokennya. Maka hasilnya akan menjadi. "perlihatkan", "nama", "mahasiswa". Proses dapat dilihat pada Gambar 2.3.



Gambar 2. 3 Tahapan *Casefolding*

5. *Remove Stopword*
Hasil penyesuaian yang dilakukan pada proses sebelumnya, akan dilakukan proses menghilangkan kata yang tidak penting seperti kata sambung dan kata pengganti orang. Proses nya dapat dilihat pada Gambar 2.4.



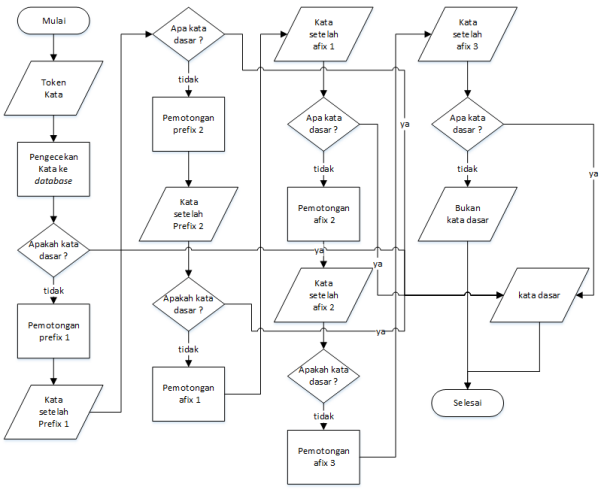
Gambar 2. 4 Tahapan *Remove Stopword*

Daftar kata yang termasuk *stopword* adalah dapat dilihat pada Gambar 2.5.

No	Kata *stopword*	No	Kata *stopword*	No	Kata *stopword*
1	Mampu	11	Karena	21	Mungkin
2	Tentang	12	Sudah	22	sebagian
3	Di	13	Oleh	23	Kata
4	Setelah	14	Bisa	24	Sering
5	Semua	15	Dia	25	Saya
6	Hampir	16	Telah	26	Belum
7	Juga	17	Jika	27	Atas
8	Ada	18	Ke	28	Sesuai
9	Seperti	19	Itu	29	Tabel
10	Jadi	20	Dalam	30	pada

Gambar 2. 5 Daftar Kata *Stopword*

6. *Stemming*
Hasil dari proses sebelumnya diambil dan akan dilakukan proses *stemming* mencari dasar kata dari setiap token. Maka akan menjadi, “lihat”, “nama”, “mahasiswa”. Proses dapat dilihat pada Gambar 2.6.



Gambar 2. 6 Tahapan *Stemming*

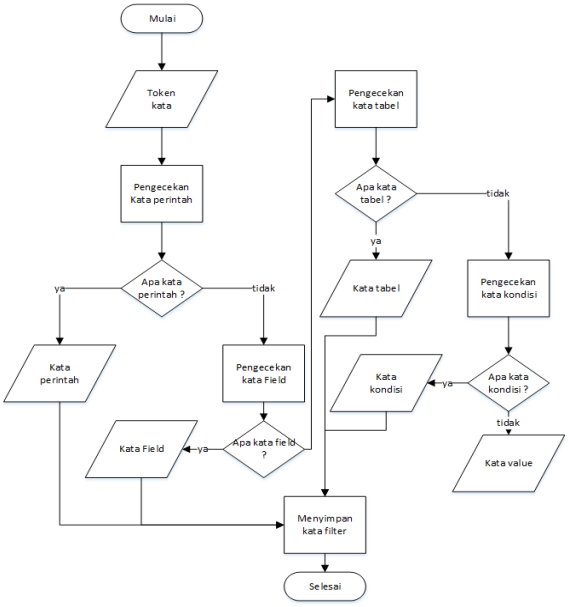
2.3 *Filtering*

Filtering merupakan suatu proses yang mengecek setiap token lalu dibandingkan dengan kamus kata, yaitu kamus kata perintah, parameter field, parameter tabel, parameter kondisi, parameter nilai. Berikut adalah tabel kata perintah dapat dilihat pada Tabel 2.1:

Tabel 2. 1 Tabel kata perintah

No	Kata Perintah
1	tampil
2	lihat
3	sebut
4	cari
5	apa

Proses dari *filtering* dapat dilihat pada flowchart pada Gambar 2.7.



Gambar 2. 7 *Flowchart Filtering*

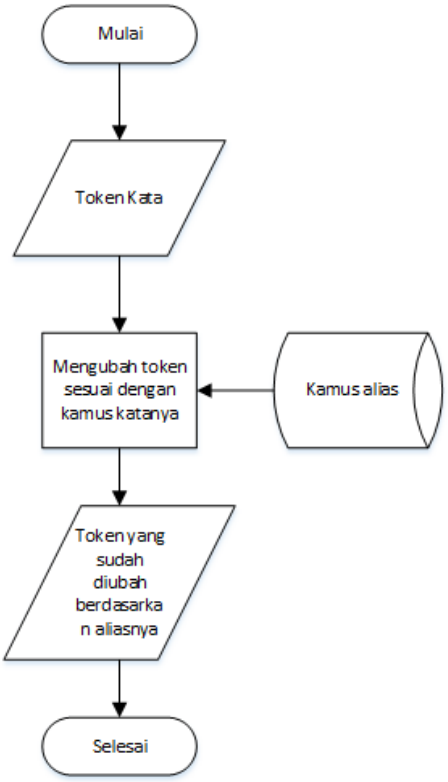
Maka akan didapatkan :

a. Kata perintah : lihat
b. Kata field : nama
c. Kata Tabel : mahasiswa

2.4 *Translating*

Setelah dilakukan analisa terhadap kalimat yang dimasukkan dan menghasilkan token yang sudah dikategorikan maka token akan diubah kedalam kata yang dimengerti query.

Proses Translating terdiri dari proses membandingkan kata yang ada pada setiap token yang telah melewati proses pengecekan dengan kamus kata yang sudah disediakan, proses nya dapat dilihat pada flowchart pada Gambar 2.8.



Gambar 2. 8 *Flowchart Translating*

Maka akan didapatan hasil dari proses translating adalah sebagai berikut :

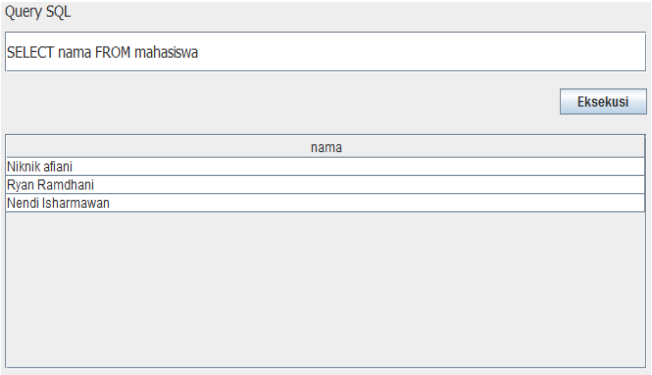
a. Kata perintah : select
b. Kata field : nama
c. Kata tabel : mahasiswa

Selanjutnya akan dilakukan penyusunan *query* sesuai dengan aturan sintaks *query*. Maka hasilnya akan menjadi :

"select nama from mahasiswa"

2.5 Eksekusi Query

Query yang telah didapatkan akan dapat dikirim ke DBMS untuk dapat diekseskusi dan mendapatkan informasi dari *query* tersebut. Hasil dapat dilihat pada Gambar 2.9.



Gambar 2. 9 Hasil Eksekusi *Query*

2.6 Hasil Pengujian

Pengujian yang dilakukan adalah pengujian jenis *query* yaitu :

a. Query tanpa kondisi

Pengujian ini dilakukan dengan menginputkan kalimat pada form masukan kemudian menekan tombol "proses". Kalimat yang dimasukan dapat dilihat pada tabel 2.2.

Tabel 2. 2 Tabel Kalimat Jenis Pertama

No	Kalimat
1	Tampilkan mahasiswa
2	Tampilkan nama mahasiswa
3	Lihat nilai uas dan uts
4	sebutkan nama, alamat dan kota mahasiswa

Query yang diharapkan dari kalimat pada Tabel 2.2 dapat dilihat pada Tabel 2.3 :

Tabel 2. 3 Tabel *Query* yang diharapkan

No	Kalimat
1	select * from mahasiswa
2	select nama from mahasiswa
3	select uas , uts from nilai
4	select nama, alamat, kota from mahasiswa

Hasil dari kalimat yang dimasukan setelah menekan tombol proses dapat dilihat pada tabel 2.4.

Tabel 2. 4 Hasil Pengujian kalimat

No Kalimat	Hasil pengujian
1	SELECT * FROM mahasiswa
2	SELECT nama FROM mahasiswa
3	SELECT uas , uts FROM nilai
4	SELECT nama , alamat , kota FROM mahasiswa

b. Query dengan kondisi

Pengujian ini dilakukan dengan menginputkan kalimat pada form masukan kemudian menekan tombol "proses". Kalimat yang dimasukan dapat dilihat pada tabel 2.5:

Tabel 2. 5 Tabel Kalimat Jenis Kedua

No	Kalimat
1	Lihat mahasiswa yang nimnya 10112322
2	tampilkan nim mahasiswa yang namanya udin
3	tampilkan nama mahasiswa yang jenis kelamin nya pria
4	tampilkan mahasiswa berdasarkan jenis kelamin

Query yang diharapkan dari kalimat pada Tabel 2.5 dapat dilihat pada Tabel 2.6.

Tabel 2. 6 Tabel *Query* yang diharapkan

No	Kalimat
1	select * from mahasiswa where nim = '10112322'
2	select nim from mahasiswa where nama = 'udin'
3	select nama from mahasiswa where jk = 'pria'
4	select nama from mahasiswa order by jk

Hasil dari kalimat yang dimasukan setelah menekan tombol proses dapat dilihat pada tabel 2.7.

Tabel 2. 7 Tabel Hasil Pengujian Kalimat

No	Kalimat
1	SELECT * FROM mahasiswa WHERE nim = '10112322'
2	SELECT nim FROM mahasiswa WHERE nama = 'udin'
3	SELECT nama FROM mahasiswa WHERE jk = 'pria'
4	SELECT * FROM mahasiswa group by jk

c. Query dengan dua kondisi

Pengujian ini dilakukan dengan menginputkan kalimat pada form masukan kemudian menekan tombol "proses". Kalimat yang dimasukan dapat dilihat pada tabel 2.8.

Tabel 2. 8 Tabel Kalimat Jenis Ketiga

No	Kalimat
1	lihat mahasiswa yang tinggal di bandung dan namanya nendi
2	tampilkan mahasiswa yang berjenis kelamin perempuan dan tinggal di bandung

Query yang diharapkan dari kalimat pada Tabel 2.8 dapat dilihat pada Tabel 2.9.

Tabel 2. 9 Tabel Query yang diharapkan

No	Kalimat
1	select * from mahasiswa where kota = 'bandung' and nama = 'nendi'
2	select * from mahasiswa where jk = 'perempuan' and kota = 'bandung'

Hasil dari kalimat yang dimasukan setelah menekan tombol proses dapat dilihat pada tabel 2.10.

Tabel 2. 10 Hasil Pengujian Kalimat

No	Kalimat
1	SELECT * FROM mahasiswa WHERE kota = 'bandung' and nama = 'nendi'
2	SELECT * FROM mahasiswa WHERE jk = 'wanita' and kota = 'bandung'

d. Query banyak tabel tanpa kondisi
Pengujian ini dilakukan dengan menginputkan kalimat pada form masukan kemudian menekan tombol "proses" banyak nya kalimat yang digunakan adalah empat dan tabel yang dipanggil harus memiliki relasi terlebih dahulu. Kalimat yang dimasukan dapat dilihat pada tabel 2.11.

Tabel 2. 11 Tabel Kalimat Jenis Keempat

No	Kalimat
1	tampilkan nilai dan kelas
2	lihat kelas mahasiswa
3	sebutkan nama mahasiswa dan nilai uas
4	sebutkan nama kelas dan nilai uas

Query yang diharapkan dari kalimat pada Tabel 2.11 dapat dilihat pada Tabel 2.12.

Tabel 2. 12 Tabel *Query* yang diharapkan

No	Kalimat
1	select * from nilai join kelas on nilai.kode_kelas = kelas.kode_kelas
2	select * from mahasiswa join kelas on mahasiswa.kode_kelas = kelas.kode_kelas
3	select nama , uas from mahasiswa join nilai on mahasiswa.nim = nilai.nim
4	select nama_kelas, uas from nilai join kelas on nilai.kode_kelas = kelas.kode_kelas

Hasil dari kalimat yang dimasukan setelah menekan tombol proses dapat dilihat pada tabel 2.13.

Tabel 2. 13 Hasil Pengujian Kalimat

No	Kalimat
1	SELECT * FROM nilai join kelas on nilai.kode_kelas = kelas.kode_kelas
2	SELECT * FROM
3	SELECT nama , uas FROM mahasiswa join nilai on mahasiswa.nim = nilai.nim
4	SELECT nama_kelas , uas FROM kelas join nilai on kelas.kode_kelas = nilai.kode_kelas

e. Query banyak tabel dengan kodisi
Pengujian ini dilakukan dengan menginputkan kalimat pada form masukan kemudian menekan tombol "proses". Kalimat yang dimasukan dapat dilihat pada tabel 2.14.

Tabel 2. 14 Tabel Kalimat Jenis Kelima

No	Kalimat
1	sebutkan mahasiswa yang nilai uasnya 80 sampai 90
2	siapa mahasiswa yang uas nya bernilai 90

Query yang diharapkan dari kalimat pada Tabel 2.14 dapat dilihat pada Tabel 2.15.

Tabel 2. 15 Tabel *Query* yang diharapkan

No	Kalimat
1	select * from mahasiswa join nilai on mahasiswa.nim = nilai.nim where uas between 80 and 90
2	select * nama from mahasiswa join nilai on mahasiswa.nim = nilai.nim where uas = '90'

Hasil dari kalimat yang dimasukan setelah menekan tombol proses dapat dilihat pada tabel 2.16.

Tabel 2. 16 Hasil Pengujian Kalimat

No	Kalimat
1	SELECT * FROM mahasiswa join nilai on mahasiswa.nim = nilai.nim WHERE uas between 80 and 90
2	SELECT * FROM mahasiswa WHERE uas = '90'

f. Query agregasi

Pengujian ini dilakukan dengan menginputkan kalimat pada form masukan kemudian menekan tombol “proses”. Kalimat yang dimasukan dapat dilihat pada tabel 2.17.

Tabel 2. 17 Tabel Kalimat jenis Keenam

No	Kalimat
1	berapa jumlah mahasiswa
2	berapa nilai uas paling tinggi
3	berapa nilai tugas paling rendah
4	berapa jumlah mahasiswa yang nilai uas lebih dari 70

Query yang diharapkan dari kalimat pada Tabel 2.17 dapat dilihat pada Tabel 2.18.

Tabel 2. 18 Tabel Query yang diharapkan

No	Kalimat
1	select count(*) from mahasiswa
2	select max(uas) from nilai
3	select min(tugas) from nilai
4	select count(*) from mahasiswa join nilai on mahasiswa.nim = nilai.nim where uas > '70'

Hasil dari kalimat yang dimasukan setelah menekan tombol proses dapat dilihat pada tabel 2.19.

Tabel 2. 19 Hasil Pengujian Kalimat

No	Kalimat
1	SELECT count(*) FROM mahasiswa
2	SELECT max(uas) FROM nilai
3	SELECT min(tugas) FROM nilai
4	SELECT count(*) FROM mahasiswa join nilai on mahasiswa.nim = nilai.nim WHERE uas = '70'

2.7 Pembahasan

Aplikasi yang dibuat mampu untuk menterjemahkan dari bahasa alami menjadi *query* sql. Hal ini dapat dibuktikan dengan pengujian yang telah dilakukan. Proses menterjemahkan mengikuti alur proses yang digunakan dalam pengolahan bahasa alami. Dimulai dengan *preprocessing, filtering, translating* hingga akhirnya pada menyusun *query*. Tampilan yang dibuat memudahkan untuk dapat melihat setiap proses yang dilakukan sehingga pengguna dapat mengerti setiap proses yang dilewati oleh aplikasi tersebut.

Dari keenam jenis *query* yang ada menghasilkan presentase keberhasilan sebesar 90,47%, karena ada beberapa kalimat yang diujikan tidak sesuai dengan harapan ataupun tidak sesuai informasi yang dihasilkannya.

Kata-kata yang tidak terdapat dalam kamus data maka tidak akan dianggap oleh aplikasi ini, sehingga perlu adanya kamus data yang besar agar dapat mengatasi berbagai *database* yang berbeda dan berbagai bidang yang berbeda. Kata yang ada dalam kamus data memiliki penyesuaian terhadap kamus data yang didapatkan dari *database* yang dituju, sehingga walaupun kamus datanya besar tetapi tidak semua kamus data akan terpakai. Perlu ada penyesuaian kamus data sehingga aplikasi tidak akan mengambil kamus data yang tidak akan terpakai.

3. PENUTUP

3.1 Kesimpulan

Setelah melakukan analisis, perancangan, dan pengujian dengan metode blackbox, diperoleh informasi bahwa sistem dapat membantu media pencari informasi karena kalimat yang menjadi masukan bisa berubah menjadi query. Hasil uji dari beberapa kalimat didapatkan keberhasilan mendapakan 90,47% menunjukan bahwa ketika kalimat yang dimasukan oleh pengguna kurang lengkap maka akan didapatkan juga query yang salah ataupun menghasilkan informasi yang salah. Hal tersebut dapat mengakibatkan kesalahan dalam mengeksekusi query. Dengan demikian media pencari informasi dapat langsung mengakses sumber informasi pada database ketika data pada database berubah, maka hasil dari data yang ditampilkan juga akan berubah.

3.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, masih perlu dilakukan beberapa kajian lebih lanjut. Adapun saran untuk penelitian lebih lanjut yaitu :

1. Menggunakan pengkoreksi kalimat ketika bertemu dengan suatu kata yang tidak lengkap ataupun salah penulisan.
2. Menggunakan data masukan berupa voice supaya pengguna tidak perlu melakukan pengetikan dalam memasukan kalimatnya.
3. Menggunakan database yang ada pada jaringan internet agar data yang diambil menjadi lebih baru..

DAFTAR PUSTAKA

[1] Aries Muslim, Kurniawan Robby. Natural Language Processing (NLP) Komunikasi Bahasa Natural Dengan Actor Virtual. Skripsi S1. Teknik Informatika. Depok: Universitas Gunadarma (2008).

[2] Wibisono Setyawan. Aplikasi Pengolahan Bahasa Alami untuk Query Basisdata Akademik

dengan Format Data XML. Skripsi S1. Teknik Informatika. Universitas Stikubank (2013).
[3] Kuspriyanto, Sujani Herry, Tjahjono Hary, Kusuma Shofyan. Pernacangan Translator Bahasa Alami Ke Dalam Format SQL. Skripsi S1. Teknik Elektro : Institut Teknologi Bandung (2005).
[4] Hartati Sri, Zuliarso Eri. Aplikasi pengolah Bahasa Alami untuk Query Basisdata XML. Yogyakarta : Universitas Gadjah Mada (2008).
[5] Naja Anwar Sariyun, Suhari Yohanes, Wibisono Setyawan. Query Bahasa Indonesia pada Basisdata Soal Ujian di Fakultas Teknologi Sistem Informasi Universitas STIKUBANK dengan Format XML. Skripsi S1. Teknik Informatika : Universitas STIKUBANK (2013).