Aplikasi Pengolah Bahasa Alami untuk Query Basisdata Akademik dengan Format Data Xml

Setyawan Wibisono
Program Studi Teknik Informatika, Universitas Stikubank
email: setyasonny@gmail.com

Abstrak

Dalam sistem informasi akademik, operator sistem adalah pengguna yang tidak mempunyai latar belakang pengetahuan query yang cukup. Sehingga kebutuhan pengguna adalah kemudahan memperoleh informasi akademik dengan menggunakan bahasa alami yaitu bahasa Indonesia.

Program aplikasi pengolah bahasa alami untuk query basisdata akademik dengan format data XML adalah program aplikasi berbasis NLP *(Natural Language Processing)*. Digunakan oleh staf administrasi akademik untuk membantu pekerjaan pencarian data dalam format XML yang berkaitan dengan data akademik mahasiswa dalam basisdata akademik tanpa harus menggunakan bahasa SQL, tetapi menggunakan bahasa Indonesia.

Masukan program aplikasi ini berupa sebuah kalimat tanya atau kalimat perintah dalam bahasa Indonesia yang sesuai dengan aturan produksi. Kalimat yang dimasukkan, akan dipilah kata yang bermakna (token) dan kata yang tidak bermakna. Token akan dibandingkan dengan aturan produksi, daftar atribut dan kondisi, untuk mendapatkan field yang dimaksudkan dalam kalimat. Dari kumpulan token akan dikonstruksikan sesuai dengan aturan produksi pembentuk kalimat untuk diterjemahkan dalam XQuery pengakses basisdata XML, sehingga menghasilkan keluaran tabel yang sesuai dengan pertanyaan atau perintah.

Dari sistem yang dihasilkan dapat disimpulkan bahwa implementasi dengan berbasis Natural Language Processing dapat digunakan sebagai alternatif dalam merancang sebuah sistem pengakses basisdata, tetapi tidak dengan menggunakan bahasa SQL. Keunggulan dari aplikasi pengolah bahasa alami ini adalah mampu menjawab query bahasa Indonesia dari implementasi 7 aturan produksi yang telah ditetapkan dengan tampilan berbentuk tabel.

Kata kunci : natural language processing, query bahasa Indonesia, token, XML, XQuery, aturan produksi

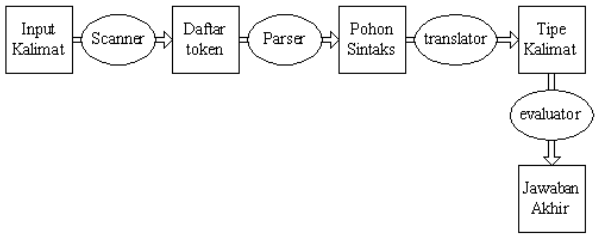
PENDAHULUAN

Bahasa adalah suatu sistem komunikasi yang mengatur tingkah laku manusia dalam bentuk ekspresi ucapan dan tulisan yang menolong dalam mengkomunikasikan perasaan dan fikiran. Dalam membentuk kata, kalimat alinea dan informasi tulisan lainnya, bahasa menggunakan suara, tanda – tanda dan simbol – simbol. Apakah ia dalam bentuk ucapan atau tulisan, bahasa merupakan media yang digunakan untuk mengkspresikan dan mengorganisasikan apa yang diketahui, dipikirkan dan dirasakan (Suparman, 1991).

Pengolahan bahasa alami secara teoritis adalah pengembangan berbagai teknik komputasi untuk menganalisis dan menampilkan teks dalam bahasa alami pada satu atau lebih tingkat analisis linguistik untuk mencapai tujuan manusia dalam hal bahasa yaitu menyelesaikan berbagai tugas atau aplikasi (Liddy, 2001).

Telah dilakukan penelitian query bahasa Indonesia untuk basisdata akademik. Sistem yang dibuat adalah dengan menggunakan bahasa alami, bahasa Indonesia untuk memberikan query masukan. Masukan yang diberikan adalah pertanyaan untuk mendapatkan informasi atau data dari basisdata (Andayani, 2002). Bahasa Indonesia sudah mempunyai *grammar* dan aturan produksi, tetapi dalam sistem ini harus ditentukan terlebih dulu aturan

produksi yang akan secara khusus menangani pola pertanyaan pada masukan. Elemen pemroses bahasa terdiri dari penganalisis leksikal, parser dan pembangkit kode atau perlakuan. Maier dan Warren dalam Hartati dan Zuliarso (Hartarti, 2008) menyatakan komponen pemrosesan bahasa alami terdiri dari scanner, parser, penterjemah (translator), optimasi query dan pengevaluator query.



Gambar 1. Komponen pengolah bahasa alami, Kaplan dalam Hartati dan Zuliarso (Hartarti, 2008)

XML kependekan dari eXtensible Markup Language, dikembangkan mulai tahun 1996 dan mendapatkan pengakuan dari W3C pada bulan Februari 1998. Seperti halnya HTML, XML juga menggunakan elemen yang ditandai dengan *tag* pembuka (diawali dengan '<' dan diakhiri dengan '>'), *tag* penutup (diawali dengan '</ 'diakhiri '>') dan atribut elemen (parameter yang dinyatakan dalam tag pembuka misal <form name="isidata">). Hanya bedanya, HTML medefinisikan dari awal *tag* dan atribut yang dipakai di dalamnya, sedangkan pada XML bisa menggunakan *tag* dan atribut sesuai kondisi (Junaedi, 2003).

Basisdata XML adalah sistem perangkat lunak yang digunakan untuk menyimpan data yang membolehkan data untuk diimpor, diakses dan diekspor dalam format XML. Basisdata XML mempunyai keunggulan lebih baik dibandingkan dengan sistem basisdata relasional jika data yang akan disimpan berupa dokumen. Dengan basisdata XML juga memungkinkan untuk melakukan penelusuran isi dokumen (Junaedi, 2003).

XML Query adalah sinonim dari XQuery. XQuery dijalankan berdasarkan ekspresi – ekpresi XPath. XQuery dan XPath memiliki model data yang sama dan mendukung fungsi – fungsi dan operator – operator yang sama. Serupa dengan XPath, XQuery didefinisikan oleh W3C dan diharapkan akan menjadi standar internasonal (Djuandi, 2008).

METODE PENELITIAN

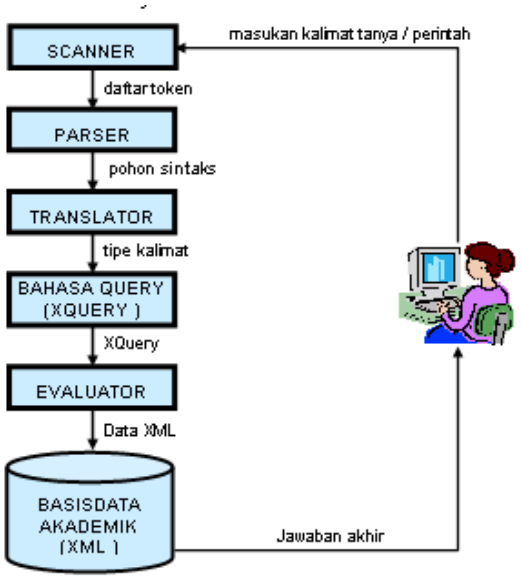
Penelitian ini merancang dan mengimplementasikan aplikasi pengolah bahasa alami untuk query basis data akademik dengan format data XML yang dapat digunakan sebagai alat bantu untuk kepentingan operasional bagian administrasi akademik dalam memperoleh informasi dari suatu basisdata akademik tanpa harus direpotkan dengan permasalahan struktur penulisan query dalam bentuk SQL standar.

Dalam membangun aplikasi ini, langkah yang dilakukan adalah melakukan studi pustaka yang berkaitan dengan sistem pengolah bahasa alami (*Natural Language Processing*). Kemudian melakukan pengamatan obyek penelitian, dalam hal ini basisdata pada sistem informasi akademik Universitas Stikubank Semarang. Hal ini bertujuan untuk mendapatkan data – data akademik yang mempunyai keterkaitan dengan topik penelitian. Dilanjutkan dengan merancang dan mengimplementasikan basisdata akademik dalam format XML. Serta merancang dan mengimplementasikan aplikasi pengolah bahasa alami dalam bahasa pemrograman.

HASIL DAN PEMBAHASAN

1. Deskripsi Sistem

Program aplikasi pengolah bahasa alami untuk query basisdata akademik dengan format data XML adalah program aplikasi yang digunakan oleh staf administrasi akademik untuk membantu pekerjaan pencarian data dalam format XML yang berkaitan dengan data akademik mahasiswa dalam basisdata akademik. Dengan menggunakan program aplikasi ini, maka staf administrasi akademik yang tidak mempunyai dasar pengetahuan SQL tetap dapat melakukan pencarian data akademik dengan menggunakan bahasa alami yaitu bahasa Indonesia.



Gambar 2. Blok diagram pengolah bahasa alami dengan data XML

Seperti terlihat pada gambar 2, masukan program aplikasi ini berupa sebuah kalimat tanya atau kalimat perintah dalam bahasa Indonesia yang diketikkan ke dalam suatu antarmuka (*form*). Bentuk kalimat tanya atau kalimat perintah harus sesuai dengan aturan produksi. Jika kalimat tanya / perintah yang dimasukkan sesuai dengan aturan produksi dan data tersedia, maka data akan ditampilkan dalam bentuk tabel. Jika kalimat tanya / perintah yang dimasukkan tidak sesuai dengan aturan produksi yang telah ditetapkan, maka hasil yang ditampilkan adalah sebuah peringatan yang menyatakan bahwa format kalimat yang dimasukkan salah. Jika kalimat tanya / perintah yang dimasukkan sesuai dengan format yang telah ditetapkan, tetapi tidak ada data yang sesuai dengan pertanyaan, maka dimunculkan sebuah tabel kosong.

Proses yang dilakukan oleh aplikasi ini adalah mengidentifikasi kata – kata yang ada pada kalimat alami dan melihat struktur kalimat. Proses awal adalah kata – kata dalam kalimat akan dibandingkan dengan aturan produksi yang telah ditetapkan dalam pembentukan kalimat. Jika memenuhi aturan dalam pembentukan kalimat, maka kalimat tersebut kemudian dibandingkan dengan daftar kata – kata yang termasuk atribut untuk mendapatkan field yang dimaksudkan dalam kalimat.

Selain dibandingkan dengan daftar atribut, kata – kata dalam bahasa alami tersebut juga dibandingkan dengan daftar kata pelengkap yang berisi daftar kata – kata alami yang dapat digunakan sebagai operator seperti kata – kata "sama dengan", "kurang dari", "lebih dari" dan lain – lain. Selanjutnya kata – kata yang termasuk kondisi operator akan didapatkan dari hasil perbandingan kata – kata alami dengan daftar kata kondisi yang berisi kata - kata keterangan kondisi seperti kata, huruf, karakter dan lain-lain.

Dari beberapa perbandingan, akan tersisa kata – kata yang tidak termasuk dalam kata – kata yang penting, sehingga kata – kata tersebut termasuk dalam kata – kata yang dapat diabaikan. Dari kumpulan kata – kata yang termasuk kata – kata penting, akan dikonstruksikan sesuai dengan aturan produksi pembentuk kalimat. Jika sudah sesuai dengan aturan produksi, maka akan diterjemahkan dalam XQuery untuk mengakses basisdata XML, sehingga menghasilkan keluaran *field* yang sesuai dengan pertanyaan atau perintah masukan. Keluaran yang dihasilkan adalah sebuah tabel yang muncul pada sebuah *form*. Tabel tersebut berisi tampilan data akademik yang sesuai dengan apa yang ditanyakan atau diperintahkan oleh pengguna.

2. Arsitektur Sistem

Aplikasi pengolah bahasa alami ini mempunyai tiga lapisan yaitu: (Junaedi, 2003)

a. Lapisan basisdata

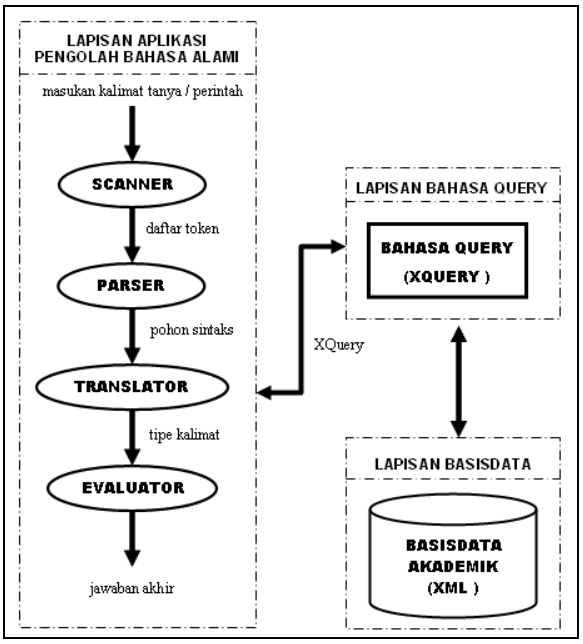
Lapisan ini digunakan untuk menyimpan dokumen XML. Dalam aplikasi ini menggunakan DBMS SQL Server 2008. Tipe data ini dapat digunakan dalam definisi tabel untuk mendefinisikan tipe sebuah kolom, tipe variabel dalam kode prosedural *Transact-SQL*, dan sebagai parameter prosedur. Kolom, variabel dan parameter dari tipe data XML dapat dibatasi dengan XML Schema. XML Schema didefinisikan dalam katalog SQL Server

b. Lapisan bahasa query

Sebagaimana dalam basisdata relasional, maka XML juga mempunyai bahasa query sendiri yang dioptimasi untuk format data. Untuk SQL Server 2008, Microsoft telah menambahkan dukungan *server-side* untuk XQuery. Berbasis pada bahasa query XPath, XQuery adalah bahasa yang dapat meng – query data XML terstruktur dan semi – terstruktur. Berpasangan dengan tipe data XML, hal ini mempercepat dan mengefisienkan penyimpanan dan temu kembali data XML.

c. Lapisan aplikasi

Lapisan ini merupakan antarmuka menggunakan masukan bahasa Indonesia, sedangkan keluaran dalam bentuk tabel. Lapisan ini berisi komponen – komponen pengolah bahasa alami. Lapisan terdiri dari scanner, parser, penterjemah (translator), optimasi query dan pengevaluator query. Bahasa pemrograman yang digunakan untuk mengimplementasikan lapisan ini adalah bahasa pemrograman Java. Pemrogaman Java menyediakan banyak fasilitas yang memudahkan untuk mengimplementasikan sistem yang dibuat.



Gambar 3. Arsitektur pengolah bahasa alami dengan data XML

Secara global sistem ini dapat digambarkan dengan arsitektur aplikasi pengolah bahasa alami untuk query bahasa Indonesia dengan format XML seperti gambar 3. Komponen aplikasi pengolah bahasa alami terdiri dari: (Junaedi, 2003)

a). Scanner

Proses scanning bertujuan mengelompokkan masukan kalimat ke dalam *token*. Scanner mengubah kalimat menjadi daftar kata yang tergolong *token* beserta data dan membuang kata – kata yang dapat diabaikan. Langkah – langkah proses *scanning* adalah sebagai berikut :

a. Membaca masukan kalimat
b. Mengubah kalimat ke dalam bentuk list kata – kata
c. Dari list kata tersebut akan dibuang karakter atau kata yang tidak berarti (dapat diabaikan).

b). Parser

Parser melakukan pelacakan masukan kalimat untuk mendapatkan langkah pembentukannya. Langkah – langkah ini dilakukan dari atas ke bawah, yaitu dari simbol awal sampai ke kalimat yang dihasilkan. Parser menganalisis sintaks daftar kata hasil *scanning* sesuai dengan aturan produksi yang ditentukan. Langkah – langkah dalam proses *scanning* adalah sebagai berikut :

1. Membaca daftar kata hasil *scanning* yang sudah terseleksi
2. Menentukan frase atribut
3. Memeriksa apakah ada ekor atribut pada frase atribut
4. Menentukan frase kondisi

Teknik yang digunakan dalam *parsing* adalah perbedaan list. Perbedaan list awal dengan ekor list akan menyisakan satu kata, kemudian kata tersebut diperiksa posisinya dalam query sesuai dengan aturan produksi untuk menentukan apakah termasuk frase atribut atau frase kondisi, apakah atribut atau konstanta.

c). Translator

Translator berfungsi untuk mengubah pohon sintak hasil *parsing* ke tipe – tipe query yang sesuai. Langkah – langkah proses penerjemahan dalam translator adalah sebagai berikut :

1. Membaca pohon sintaks hasil *parsing*.
2. Menempatkan atribut dari frase atribut ke posisi atribut pertama dalam notasi tipe query.
3. Menempatkan atribut dan konstanta (data, int, opr) dari frase kondisi ke dalam notasi tipe query sesuai posisinya.

Aturan produksi yang ditentukan dapat menghasilkan kalimat yang kompleks, karena frase atribut dapat memiliki satu ekor atribut dan frase kondisi dapat mempunyai nol atau satu ekor frase kondisi. Akan tetapi, memperhatikan pertanyaan yang biasa digunakan untuk mengakses data dari basisdata akademik, maka translator hanya dirancang untuk mengubah pohon sintaks yang susunan frase atribut dan frase kondisinya sesuai dengan pola pertanyaan – pertanyaan.

d). Evaluator

Evaluator berfungsi menentukan jawaban akhir query berdasarkan hasil keluaran translator yang berupa penggolongan query menurut tipenya dengan urutan atribut sesuai dengan notasi masing – masing tipe query. Proses yang terjadi dalam evaluator adalah pencarian relasi yang terdapat dalam query dan pencarian nilai yang tepat dalam basisdata sesuai dengan relasi tersebut. Setiap tipe query mempunyai urutan atribut yang berbeda yang akan menentukan proses evaluasinya.

3. Aturan Produksi

Aturan produksi yang dibuat adalah aturan produksi yang dapat mengakses data karena pola aturan produksi yang sudah ditentukan adalah pola aturan produksi yang sesuai dengan pola query pada basisdata dengan format data XML.

Untuk merancang aturan produksi, dapat dilakukan dengan menentukan lebih dahulu pola keteraturan pertanyaan untuk mengakses basisdata. Pola keteraturan aturan produksi yang tepat ditampilkan dalam contoh pertanyaan dan pernyaan sebagai berikut :

1. Apa mata kuliah yang ditawarkan ?
2. Tampilkan daftar dosen
3. Apa mata kuliah dan siapa dosen pengajar ?
4. Siapa dosen pengajar mata kuliah Jaringan Syaraf Tiruan ?
5. Tampilkan nim dan nama dengan program studi teknik informatika
6. Tampilkan nama mahasiswa peserta mata kuliah Pemrograman Web

Dari contoh pertanyaan dan pernyataan, dapat diidentifikasi bahwa pertanyaan dan pernyataan tersebut terdiri dari dua frase, yaitu :

a. Frase yang ditanyakan, atau disebut juga sebagai frase atribut. Disebut sebagai frase atribut sebab data yang ditanyakan adalah salah satu nama atribut dari tabel yang terdapat dalam basisdata. Frase atribut dapat terdiri dari satu atribut, misalnya nim, nama, mata kuliah atau dua atribut seperti program studi dan tahun masuk.
b. Frase yang diketahui, atau disebut juga sebagai frase kondisi. Disebut sebagai frase kondisi sebab berisi atribut dan data yang memberi batasan kondisi pada bagian yang ditanyakan. Frase kondisi dapat terdiri dari satu kondisi, misalnya "mahasiswa = Nur", "sebelum tahun 2005", atau dua kondisi "nama mahasiswa dengan tahun masuk sebelum tahun 2005 dan program studi teknik informatika".

Berdasarkan pola keteraturan di atas, maka aturan produksi dengan simbol awal <query> ditentukan seperti di bawah ini :

S → <atribut>A

A → <kata_sambung>B | **ε** | <atribut_kondisi>C

B → <atribut>A

C → <operator_bukan>D | <data>E | <operator>F

D → <data>E

F → <data>E

E → <kata_sambung>G | **ε**

G → <atribut_kondisi>C

<atribut> → nim|nama|jeniskelamin|alamat|
notelepon|tempatlahir|tanggallahir|
programstudi|ip|masastudi|status|
tahunmasuk|skkelulusan|
judulskripsi|pembimbing|kode|
matakuliah|semester|sks|nilai|dosen

<kata_sambung> → ‘,’| dan

<atribut_kondisi> → ’sebelum’|’sesudah’

<operator_bukan> → ‘bukan’|’tidak’

<operator> → ‘lebih|’kurang’

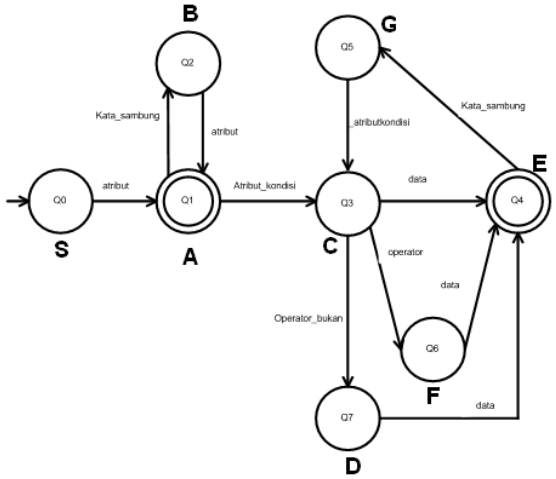
<data> → tergantung dari data yang ada pada query

Arti notasi yang digunakan dalam aturan produksi tersebut

→ : didefinisikan sebagai

< > : simbol non terminal

Bila kumpulan aturan produksi ditampilkan dalam bentuk Finite State Automata, maka akan seperti terlihat pada gambar 4.



Gambar 4. Finite state automata

4. Kata – Kata Yang Diabaikan

Dari beberapa contoh pertanyaan, dapat terlihat bahwa penentuan bentuk query dalam bahasa Indonesia akan sangat berpengaruh dalam proses penentuan jawaban dari pertanyaan tersebut.

Pertanyaan dan pernyataan untuk melakukan query dalam bahasa Indonesia tersusun dari sekumpulan kata yang dapat dikategorikan sebagai kalimat tanya atau kalimat perintah. Kata yang digunakan untuk mengawali pertanyaan adalah kata tanya, sedangkan kata yang digunakan untuk mengawali pernyataan adalah kata perintah. Susunan kalimat terdiri dari kata tanya atau kata perintah, kata – kata yang bermakna (token), konstanta atau data dan kata – kata yang hanya berfungsi sebagai kata pelengkap untuk kalimat. Kata pelengkap dapat berupa kata keterangan, kata depan dan kata kerja.

Dalam sebuah kalimat untuk melakukan query bahasa Indonesia, kata tanya atau kata perintah yang digunakan untuk mengawali masukan akan diabaikan dalam proses penentuan jawaban. Begitu pula untuk kata yang berfungsi sebagai pelengkap kalimat / pertanyaan juga akan diabaikan dalam proses penentuan jawaban. Sehingga dapat diidentifikasikan kata apa saja yang diabaikan. Pada tabel 1. diperlihatkan kata – kata yang dapat diabaikan.

KATA TANYA	KATA PERINTAH	KATA PELENGKAP	KATA KETERANGAN
siapa	tampilkan	saja	di
dimana	tampil	daftar	pada
apa	cari	ditawarkan	dari
apa saja		pengajar	yang
		mengandung	oleh
		diambil	dengan
		mahasiswa	

Tabel 1. Daftar kata yang dapat diabaikan

Berdasarkan aturan produksi yang telah ditentukan, maka kata –kata yang mempunyai arti (token) yang terdapat dalam aplikasi ini antara lain atribut, operator, kata sambung, bukan, data. Daftar kata – kata yang termasuk token ditunjukkan dalam tabel 2.

Tabel 2. Daftar kata yang termasuk token

ATRIBUT	OPERATOR TAHUN	KATA SAMBUNG	OPERATOR BUKAN
Nim	sebelum	dan	bukan
Nama	sesudah	,	tidak
Jeniskelamin	urang		selain
Alamat	lebih		
Notelepon			
Tempatlahir			
Tanggallahir			
Programstudi			
Ip			
Masastudi			
Status			
Tahunmasuk			
Tahunlulus			
Skkelulusan			
Judulskripsi			
Pembimbing			
Matakuliah			
Kode			
Matakuliah			
Semester			
Sks			
Nilai			
dosen			

Dari hasil pemecahan kalimat, maka terdapat kategori kata yang termasuk token, kategori kata yang tidak termasuk token dan kata yang tidak termasuk kategori sebagai token tetapi juga kata yang tidak termasuk sebagai kata yang dapat diabaikan, kata – kata tersebut akan dianggap sebagai data.

5. Tipe Query

Aplikasi pengolah bahasa alami ini mampu menampilkan hasil query dalam banyak query bahasa indonesia, tetapi harus sesuai dengan salah satu dari 7 aturan produksi yang ditetapkan. Maka perlu dirancang tipe query yang dapat diimplementasikan dalam aplikasi ini. Berdasarkan identifikasi yang telah dilakukan Andayani (2002), terdapat tujuh tipe query sebagai berikut :

1. Tipe q_a (query – atribut)

   Tipe query ini hanya berisi satu atribut pada kalimat yang berfungsi sebagai pertanyaan atau pernyataan. Atribut itulah yang akan ditampilkan. Query ini merupakan tipe yang paling sederhana karena yang hanya memuat atribut yang ditanyakan.

2. Tipe q_a_a (query – atribut – atribut)

   Tipe query ini berisi beberapa atribut yang akan ditampilkan. Untuk memisahkan satu atribut dengan atribut berikutnya digunakan kata sambung ‘dan’ atau tanda baca koma ‘,’.

3. Tipe q_a_opr (query – atribut – operator)

   Tipe query berisi satu atribut yang akan ditampilkan dan mempunyai satu kondisi.

4. Tipe q_a_a_opr (query – atribut – atribut – operator – atribut – operator)

   Tipe query berisi beberapa atribut yang akan ditampilkan dan beberapa kondisi. Tipe query ini memuat beberapa atribut yang ditanyakan. Untuk memisahkan satu atribut dengan atribut berikutnya digunakan kata sambung ‘dan’ atau tanda baca koma ‘,’. Demikian juga untuk memisahkan satu atribut kondisi dengan atribut kondisi berikutnya digunakan kata sambung ‘dan’ atau tanda baca koma ‘,’.

5. Tipe q_a_opr_data (query – atribut – operator – < data>)

   Tipe query berisi beberapa atribut yang akan ditampilkan dan kondisi operator ‘lebih’, ‘kurang’, ‘sebelum’ atau ‘sesudah’.

6. Tipe q_a_bukan (query – atribut – bukan – data)

   Tipe query berisi sebuah atribut yang akan ditampilkan dan kondisi operator “bukan” atau “tidak” atau “selain”.

7. Tipe q_a_a_bukan (query – atribut – atribut – bukan – data)

   Tipe query berisi beberapa atribut yang akan ditampilkan dan kondisi operator “bukan” atau “tidak” atau “selain”.

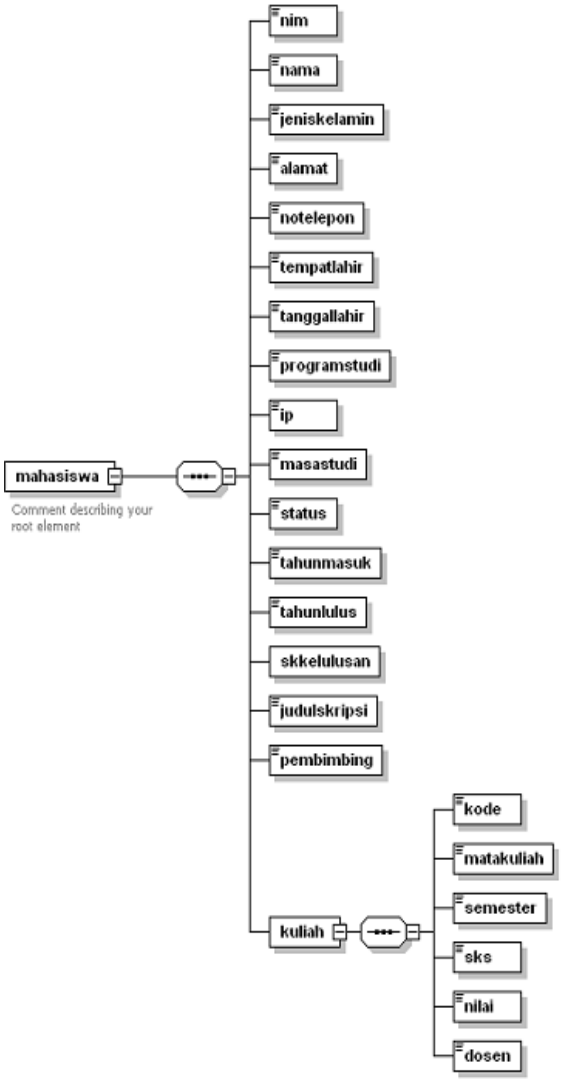
6. Struktur Data

Dalam penyimpanan data dengan format XML, maka data akademik dibuat dalam XML schema. Pembuatan XML Schema dimaksudkan untuk memastikan bahwa elemen – elemen dan atribut – atribut yang dimasukkan ke dalam dokumen sudah memenuhi aturan yang diterapkan dalam skema. XML Schema

tabel_akademik2 untuk data yang disimpan adalah sebagai berikut :

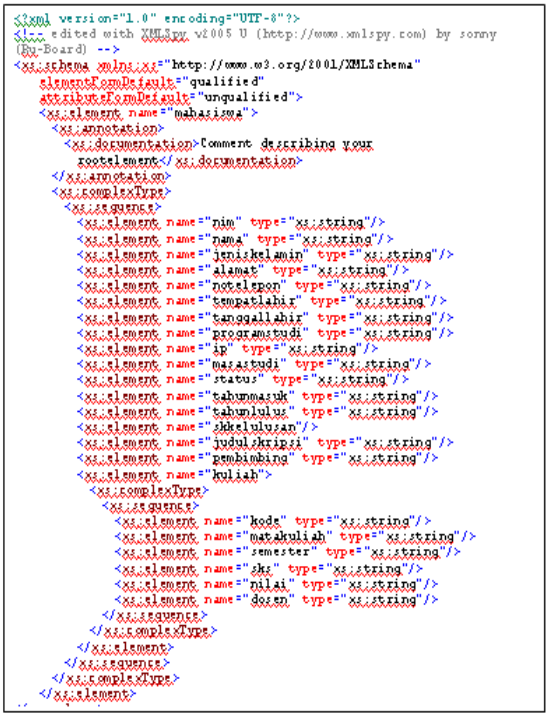
1. Nim
2. Nama
3. Jenis Kelamin
4. Alamat
5. No Telepon
6. Tempat Lahir
7. Tanggal Lahir
8. Program Studi
9. IP
10. Masa Studi
11. Status
12. Tahun Masuk
13. Tahun Lulus
14. SK Kelulusan
15. Judul Skripsi
16. Pembimbing
17. Mata Kuliah :
    a. Kode Mata Kuliah
    b. Nama Mata Kuliah
    c. Semester
    d. SKS
    e. Nilai
    f. Nama Dosen

Penggambaran data dengan bentuk diagram akan terlihat seperti gambar 5. Field yang dimiliki mahasiswa akan diperlihatkan sebagai field utama, sedangkan field yang dimiliki oleh kuliah diperlihatkan sebagai sebuah susunan yang bersarang.



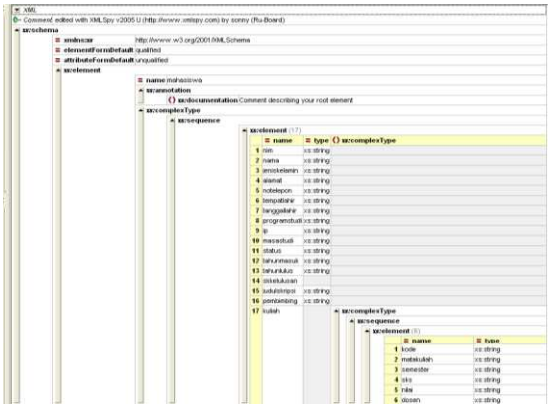
Gambar 5. Diagram XML

Penggambaran data dengan format XML pada XML Schema akan terlihat seperti gambar 6. Field yang dimiliki mahasiswa akan diperlihatkan sebagai field utama dengan tipe data string, sedangkan field yang dimiliki oleh kuliah diperlihatkan sebagai sebuah susunan yang bersarang dengan tipe data string.



Gambar 6. XML Schema

Penggambaran data dengan format XML pada XML Grid akan terlihat seperti gambar 7. Field yang dimiliki mahasiswa akan diperlihatkan sebagai elemen utama, sedangkan field yang dimiliki oleh kuliah diperlihatkan sebagai elemen dengan susunan yang bersarang.



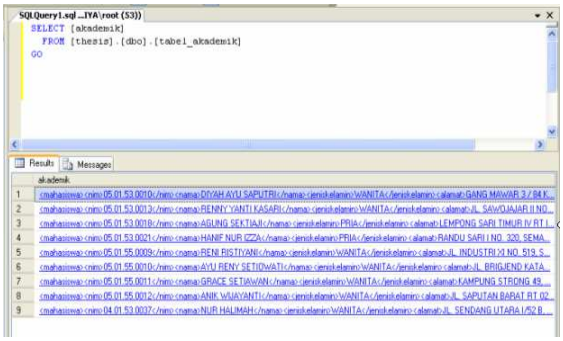
Gambar 7. XML Grid

7. Implementasi Basisdata

Basisdata yang digunakan dalam aplikasi ini adalah basisdata akademik Fakultas Teknologi Informasi, Universitas Stikubank Semarang dalam format XML. Dalam implementasi basisdata ini yang digunakan adalah data mahasiswa, yang meliputi data dari 5 program studi dalam Fakultas Teknologi Informasi. Berikut ini adalah salah satu contoh data XML yang diimplementasikan dalam sebuah tabel dengan nama tabel_akademik2. Tabel tersebut ditampil dalam struktur data seperti ditampilkan dalam gambar 8.

Gambar 8. Struktur data akademik dalam format XML

Pembuatan data akademik diimplementasikan menggunakan SQL Server 2008 dalam format XML. Gambar 9 adalah tampilan data akademik dalam format XML.



Gambar 9. Tampilan data akademik dalam format XML

8. Antarmuka

Implementasi aplikasi pengolah bahasa alami untuk query basisdata akademik dengan format data XML menyajikan suatu perangkat lunak aplikasi yang dapat digunakan sebagai alat bantu untuk kepentingan operasional bagian administrasi akademik dalam memperoleh informasi dari suatu basisdata akademik tanpa harus direpotkan dengan permasalahan struktur penulisan query dalam bentuk SQL standar.

Antarmuka aplikasi pengolah bahasa alami untuk query basisdata akademik dengan format XML dibuat dalam bentuk sederhana, seperti terlihat pada gambar 10, hanya terdiri dari empat bagian yaitu :

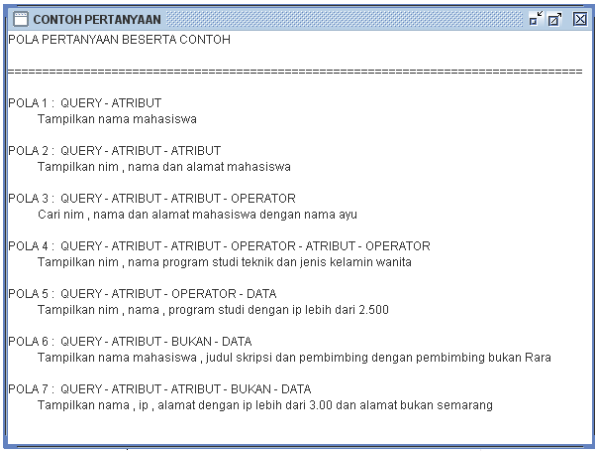
1. Bagian untuk memasukkan query dalam bentuk kalimat tanya atau kalimat perintah.
2. Bagian tombol **"RUN QUERY !!!"** untuk memerintahkan untuk melakukan pemrosesan terhadap query
3. Bagian untuk menampilkan hasil query dalam bentuk tabel,
4. Bagian tombol **"LIHAT CONTOH"** untuk menampilkan contoh yang berisi aturan produksi dan contoh pola pertanyaan yang dapat dijadikan sebagai query.

Gambar 10. Tampilan awal progam

Pada saat aplikasi dijalankan pertama kali maka akan langsung muncul contoh query yaitu **Tampilkan Nama Mahasiswa** dan tampilan hasil query dalam tabel hasil. Selanjutnya pemakai memasukkan query yang diinginkan dan kemudian menekan tombol **" RUN QUERY !!! "**, maka hasilnya akan ditampilkan di bagian tabel hasil. Bila data penulisan query tidak sesuai dengan aturan produksi, maka akan ditampilkan pesan kesalahan, seperti terlihat pada gambar 11.

Gambar 11. Tampilan pesan kesalahan

Bila menginginkan untuk terlebih dahulu melihat pola pertanyaan dan contoh pertanyaan yang dapat menghasilkan jawaban, maka bisa menekan tombol **"LIHAT CONTOH ",** dengan tampilan seperti gambar 12.



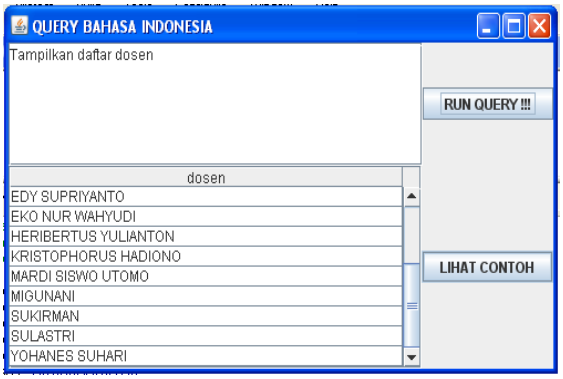
Gambar 12. Tampilan pola dan contoh pertanyaan

9. Hasil Pengolahan Query

Dari aturan produksi yang telah ditetapkan maka sistem hanya bisa menerima masukan yang sesuai. Proses awal yang dilakukan terhadap kalimat tersebut adalah pembentukkan daftar token yang dilakukan oleh scanner. Token – token ini akan diproses oleh parser. Parser melakukan pelacakan terhadap pembentukan kalimat yang kemudian dianalisa kesesuaiannya dengan aturan produksi yang ada. Penterjemahan kalimat hasil dari pohon sintaks dilakukan oleh translator yang menghasilkan tipe kalimat. Dalam proses ini akan diketahui apakah kalimat perintah yang dimasukkan itu sesuai dengan aturan produksi yang ditetapkan atau tidak untuk mendapatkan jawaban akhir yang diinginkan pengguna. Bila sesuai, maka tipe kalimat diproses oleh evaluator untuk mendapatkan hasil operasi query.

Hasil pengujian disajikan untuk melihat hasil operasi query pada basisdata XML. Dalam hasil uji coba ditampilkan masukan dalam bahasa Indonesia, hasil evaluasi XQuery pada basisdata XML pada SQL Server dan statemen XQuery. Salah satu pengujian dengan menggunakan aturan produksi paling sederhana, yaitu aturan produksi tipe q – a (query – atribut).

1. Pengujian aturan produksi tipe (query – atribut) Pengujian dengan contoh query masukan : "**Tampilkan daftar dosen".** Tipe query ini adalah query untuk meminta data yang berupa sebuah atribut dari tabel. Query ini meminta aplikasi untuk menampilkan semua dosen yang terdapat dalam tabel. Hasil yang ditampilkan seperti pada gambar 13, akan sesuai dengan permintaan yaitu sebuah tampilan berupa tabel yang berisi daftar semua dosen yang terdapat dalam tabel.



Gambar 13. Pengujian 1, tipe (query – atribut)

Hasil setiap tahap pemrosesan untuk query tersebut sebagai berikut (Andayani, 2002) :

1. Scanner : ["dosen"]

   Masukan query diubah menjadi list kata – kata, yaitu tampilkan, daftar dan dosen. Kemudian kata "tampilkan" dan "daftar" dibuang karena termasuk dalam kata – kata yang dapat diabaikan, sehingga diperoleh kata "alamat" sebagai kata yang bermakna (token).

2. Parser : query(fraseatribut("dosen",kosong))

   Hasil scanner diterima oleh parser untuk dianalisis sintaksnya berdasarkan aturan produksi. Sesuai dengan sintaks query, maka sintaks query tipe ini meminta sebuah atribut dengan frase kondisi yang kosong. Atau dapat juga dikatakan bahwa tipe query ini hanya meminta sebuah atribut tanpa ada kondisi yang mengikutinya.

3. Translator : q_a("dosen")

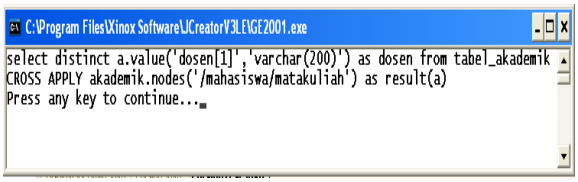
   Hasil parser diolah oleh translator dan terjadi proses pemadanan atribut dari pohon sintaks untuk memperoleh tipe query yang sesuai. Pada proses ini ada pencarian nama atribut dari data yang diketahui, dan diperoleh atribut "dosen".

4. Evaluator

   Proses yang terjadi dalam evaluator hingga diperoleh jawaban akhir, diawali dengan penentuan nama tabel dari atribut yang ditanyakan. Data yang diketahui harus dicari data lengkapnya, yaitu semua dosen. Dalam proses pencarian data lengkap ini ada

pengecekan apakah kata sebagai representasi atribut yang diketahui perlu dicari kata lengkapnya atau tidak dengan berdasarkan nama atributnya. Sebab ada beberapa kata yang merupakan representasi dari atribut, dalam masukan kalimat disebutkan sebagai 2 kata yang terpisah oleh spasi. Sehingga atribut dengan bentuk tersebut, perlu dicari kata lengkap sehingga dapat cocok dengan nama field dalam tabel.

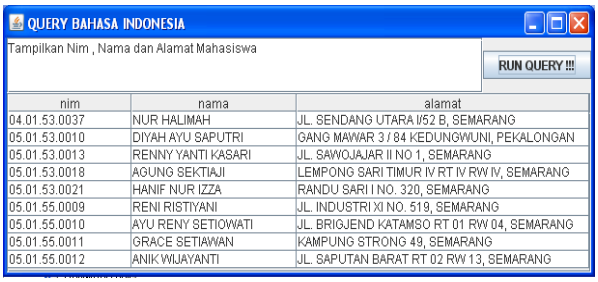
Dari keseluruhan proses, maka didapatkan hasil penerjemahan dari masukan yang berupa kalimat tanya / perintah menjadi suatu pernyataan XQuery, seperti terlihat pada gambar 14. Pernyataan XQuery tersebut kemudian dikoneksikan dengan tabel di database akademik.



Gambar 14. Translasi query tampilkan daftar dosen

2. Peyugujian aturan produksi (query – atribut – atribut), Query masukan : **Tampilkan Nim , Nama dan Alamat Mahasiswa**

   Tipe query ini adalah query untuk meminta data yang berupa beberapa atribut sekaligus dari tabel. Atribut yang hendak ditampilkan dapat terdiri dari 2 atribut atau lebih. Pada contoh query ini meminta aplikasi untuk menampilkan data Nim, Nama dan Alamat dari mahasiswa yang terdapat dalam tabel. Hasil yang ditampilkan seperti gambar 15, akan sesuai dengan permintaan yaitu sebuah tampilan berupa tabel yang berisi semua Nim, Nama dan Alamat mahasiswa yang terdapat dalam tabel dengan nama tabel_akademik2.



nim	nama	alamat
04.01.53.0037	NUR HALIMAH	JL. SENDANG UTARA I/52 B, SEMARANG
05.01.53.0010	DIYAH AYU SAPUTRI	GANG MAWAR 3 / 84 KEDUNGWUNI, PEKALONGAN
05.01.53.0013	RENNY YANTI KASARI	JL. SAWOJAJAR II NO 1, SEMARANG
05.01.53.0018	AGUNG SEKTIAJI	LEMPONG SARI TIMUR IV RT IV RW IV, SEMARANG
05.01.53.0021	HANIF NUR IZZA	RANDU SARI I NO. 320, SEMARANG
05.01.55.0009	RENI RISTIYANI	JL. INDUSTRI XI NO. 519, SEMARANG
05.01.55.0010	AYU RENY SETIOWATI	JL. BRIGJEND KATAMSO RT 01 RW 04, SEMARANG
05.01.55.0011	GRACE SETIAWAN	KAMPUNG STRONG 49, SEMARANG
05.01.55.0012	ANIK WIJAYANTI	JL. SAPUTAN BARAT RT 02 RW 13, SEMARANG

Gambar 15. Pengujian 2, tipe q_aa

Hasil setiap tahap pemrosesan untuk query tersebut sebagai berikut :

1. Scanner : [“nim”, ”,” ,”nama”, “dan”, ”alamat”]

   Masukan query diubah menjdi list kata – kata, yaitu tampilkan, nim, nama, alamat. Kemudian kata “tampilkan” dibuang karena termasuk dalam kata – kata yang dapat diabaikan. Kata sambung “dan” serta tanda “koma” akan dimengerti oleh program sebagai kata sambung, sehingga diperoleh kata “nim”, ”nama”, “alamat” sebagai kata yang bermakna (token),.

2. Parser : query(fraseatribut(“nim”, ”nama”, “alamat” , kosong))

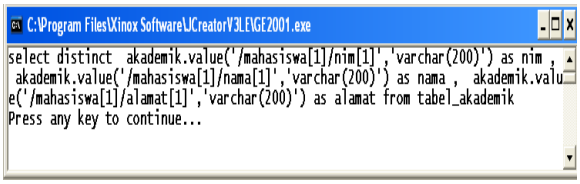
   Hasil scanner diterima oleh parser untuk dianalisis sintaksnya berdasarkan aturan produksi. Sesuai dengan sintaks query, maka sintaks query tipe ini meminta atribut “nim”, “nama”, “alamat” dengan frase kondisi yang kosong.

3. Translator : q_aa(“nim”, “nama”, “alamat”)

   Hasil parser diolah oleh translator dan terjadi proses pemadanan atribut dari pohon sintaks untuk memperoleh tipe query yang sesuai. Pada proses ini ada pencarian nama atribut dari data yang diketahui, dan diperoleh atribut “nim”, “nama”, “alamat”.

4. Evaluator

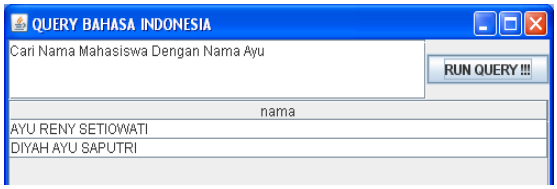
   Dari keseluruhan proses, maka didapatkan hasil penerjemahan dari masukan yang berupa kalimat tanya / perintah menjadi suatu pernyataan XQuery, seperti terlihat pada gambar 16. Pernyataan XQuery tersebut kemudian dikoneksikan dengan tabel di database akademik .



Gambar 16. Translasi query tampilkan nim , nama dan alamat mahasiswa

3. Pengujian 3, tipe q_a_opr (query – atribut – atribut – operator), Query masukan : **Cari Nama Mahasiswa dengan Nama Ayu**

   Tipe query ini adalah query untuk meminta data yang berupa beberapa atribut sekaligus dari tabel. Atribut yang hendak ditampilkan dapat terdiri dari 2 atribut atau lebih dengan sebuah kondisi yang diketahui. Pada contoh query ini meminta aplikasi untuk menampilkan data mahasiswa dengan nama yang mengandung sebuah kondisi yang diketahui ayu. Hasil yang ditampilkan seperti gambar 17, akan sesuai dengan permintaan yaitu sebuah tampilan berupa tabel yang berisi semua Nama mahasiswa yang mengandung nama ayu.



Gambar 17. Pengujian 3, tipe q_a_opr (query – atribut – atribut – operator)

Hasil setiap tahap pemrosesan untuk query tersebut sebagai berikut :

1. Scanner : [“nama”, “nama”, ”ayu”]

   Masukan query diubah menjdi list kata – kata, yaitu cari, nama, alamat. Kemudian kata “cari” dibuang karena termasuk dalam kata – kata yang dapat diabaikan. Diperoleh kata ”nama”, sebagai kata yang bermakna (token), serta aribut “nama” berikutnya sebagai kondisi, dan “ayu” sebagai data.

2. Parser :

   query(fraseatribut(“nama”,kosong),fkondisi(“nama”,data(“ayu”, kosong)))

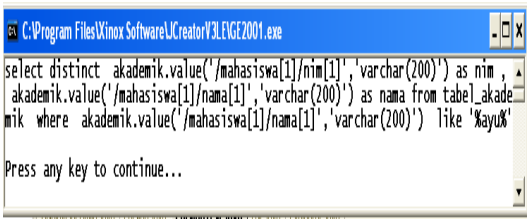
   Hasil scanner diterima oleh parser untuk dianalisis sintaksnya berdasarkan aturan produksi. Sesuai dengan sintaks query, maka sintaks query tipe ini meminta atribut “nama” dengan frase kondisi “nama” dengan data “ayu”.

3. Translator : q_a_opr(“nama”, “nama”, “ayu”)

   Hasil parser diolah oleh translator dan terjadi proses pemadanan atribut dari pohon sintaks untuk memperoleh tipe query yang sesuai. Pada proses ini ada pencarian atribut “nama” dari data yang diketahui “ayu”

4. Evaluator

   Dari keseluruhan proses, maka didapatkan hasil penerjemahan dari masukan yang berupa kalimat tanya / perintah menjadi suatu pernyataan XQuery, seperti terlihat pada gambar 18. Pernyataan XQuery tersebut kemudian dikoneksikan dengan tabel di database akademik melalui ResultSetTableModel.



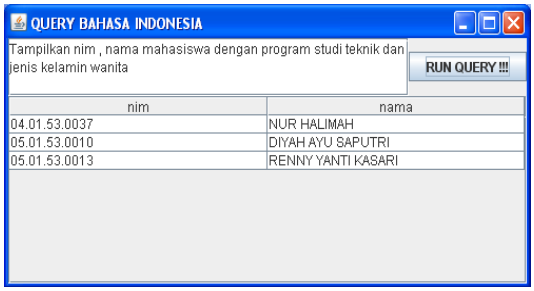
Gambar 18. Translasi query cari nama mahasiswa dengan nama ayu

4. Pengujian 4, tipe q_aa_opr

   Query masukan : **Tampilkan nim , nama mahasiswa dengan program studi teknik dan jenis kelamin wanita**

   Tipe query ini adalah query untuk meminta data yang berupa beberapa atribut sekaligus dari tabel. Atribut yang hendak ditampilkan dapat terdiri dari 2 atribut atau lebih dengan kondisi yang diketahui dua buah atau lebih. Pada contoh query ini meminta aplikasi untuk menampilkan data mahasiswa dengan nama yang mengandung dua buah kondisi yang diketahui. Hasil yang ditampilkan seperti gambar 19, akan sesuai dengan permintaan yaitu sebuah tampilan berupa

tabel yang berisi semua nim dan nama mahasiswa yang mengandung dengan program studi teknik dan jenis kelamin wanita.



Gambar 19. Pengujian 4, tipe q_aa_opr

Hasil setiap tahap pemrosesan untuk query tersebut sebagai berikut :

1. Scanner : [“nim”, “nama”, “program studi“, ”teknik”, ”jenis kelamin”, “wanita”]

   Masukan query diubah menjadi list kata – kata, yaitu tampilkan, nim, nama, program studi, teknik, jenis kelamin, wanita. Kemudian kata “tampilkan”, “dengan” dibuang karena termasuk dalam kata – kata yang dapat diabaikan. Diperoleh kata “nim”, “nama”, “program studi“, ”teknik”, ”jenis kelamin”, “wanita” sebagai kata yang bermakna (token), serta aribut “nama” berikutnya sebagai kondisi, dan “ayu” sebagai data.

2. Parser : fraseatribut(atribut, kosong), fkondisi(atribut, data, fkondisi(atribut, data, kosong))

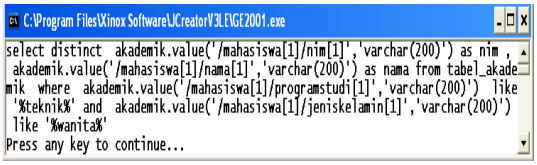
   query(“nim”,”nama”,kosong),fkondisi(“program studi”,”teknik”, fkondisi(“jenis kelamin”, “wanita”, kosong)))

   Hasil scanner diterima oleh parser untuk dianalisis sintaksnya berdasarkan aturan produksi. Sesuai dengan sintaks query, maka sintaks query tipe ini meminta atribut “nim” dan “nama” dengan frase kondisi “program studi” dengan data “teknik” dan frase kondisi “jenis kelamin” dengan data “wanita”.

3. Translator : q_a_a_opr(“nim”,”nama”, “program studi”, ”teknik”, “jenis kelamin”, “wanita”)

   Hasil parser diolah oleh translator dan terjadi proses pemadanan atribut dari pohon sintaks untuk memperoleh tipe query yang sesuai. Pada proses ini ada pencarian atribut “nim” dan “nama” dari data “program studi” yang diketahui “teknik” dan data “jenis kelamin” yang diketahui “wanita”.

4. Evaluator : dari keseluruhan proses, maka didapatkan hasil penerjemahan dari masukan yang berupa kalimat tanya / perintah menjadi suatu pernyataan XQuery, seperti terlihat pada gambar 20. Pernyataan XQuery tersebut kemudian dikoneksikan dengan tabel di database akademik melalui ResultSetTableModel.



Gambar 20. Translasi query tipe q_aa_opr

KESIMPULAN

Dari hasil penelitian yang telah dilakukan dapat disimpulkan hal – hal sebagai berikut:

1. Aplikasi ini mampu memberikan informasi akademik dengan format data XML dari sebuah permintaan berupa masukan dalam bahasa Indonesia.
2. Terdapat 7 aturan produksi yang dapat diimplementasikan dalam query bahasa Indonesia yang mampu dijawab oleh aplikasi ini.
3. Jika terdapat kesalahan dalam memberikan masukan query, maka aplikasi ini mampu memberikan peringatan terhadap kesalahan masukan query. Tetapi aplikasi ini belum mampu memberikan koreksi otomatis sebagai alternatif perbaikan kesalahan.
4. Aplikasi ini mampu memberikan keluaran berupa tabel yang berisi data. Aplikasi ini juga mampu menampilkan XQuery hasil translasi dari kalimat masukan, dalam hal ini berupa pernyataan SQL dalam format XML.

DAFTAR PUSTAKA

Andayani, S., 2002, Query Bahasa Indonesia untuk Basis Data Akademik, Tesis

Magister Komputer, Program Pasca Sarjana Ilmu Komputer, UGM, Yogyakarta.

Djuandi, F., 2008, Jurus Baru Pemrograman SQL Server 2005, Elex Media Komputindo, Jakarta.

Hartati, S., dan Zuliarso E., 2008, Aplikasi Pengolah Bahasa Alami untuk Query Basisdata XML, Dinamik, Vol XIII, No 2, Juli 2008, 168 – 175.

Junaedi, M., 2003, Pengantar XML, www.Ilmukomputer.com/umum/junaedi-xml.php, diakses tanggal : 9 Februari 2009

Liddy, E.D., 2001, Natural Language Processing, In Encyclopedia of Library and Information Science, 2nd Edition, Marcel Decker Inc, NY, USA.

Suparman, 1991, *Mengenal Artificial Intelligence*, Andi Offset, Yogyakarta.