NATURAL LANGUAGE PROCESSING DALAM MEMPEROLEH INFORMASI AKADEMIK MAHASISWA UNIVERSITAS ATMA JAYA MAKASSAR

Erick Alfons Lisangan
Fakultas Teknologi Informasi, Universitas Atma Jaya Makassar
erick_lisangan@lecturer.uajm.ac.id

ABSTRACT

The information has become one of the most important parts for an institution today, including Universitas Atma Jaya Makassar. One of the information is academic, so it needs fast processing to get that information. The problems that arise are difference of language between the user who need information and database language which storage data of academic. So we need an interpreter to bridge the difference of language. In this research, the writer uses Natural Language Processing as the solution from that problem and use Parser Noise-Disposal. The application is designed by PHP as programming language and MySQL as database.

Keywords: *natural language processing, parser noise-disposal, information*

1. PENDAHULUAN

Informasi telah menjadi salah satu bagian yang penting dari suatu institusi, termasuk dalam hal ini institusi yang bergerak di bidang pendidikan. Universitas Atma Jaya Makassar (UAJM) sebagai salah satu institusi dalam bidang pendidikan tentu membutuhkan sejumlah informasi, seperti informasi akademik, keuangan, dan lainnya. Informasi tersebut yang akan digunakan sebagai pegangan dalam pengambilan keputusan oleh *top level management*. Informasi akademik dapat berhubungan dengan mahasiswa, dosen, hingga mata kuliah. Pemakai umum terhadap informasi akademik adalah Ketua Program Studi, Dekan Fakultas, Biro Akademik, Wakil Rektor, hingga Rektor.

Informasi akademik di UAJM dapat diperoleh secara langsung dari Sistem Informasi Akademik (SIAMIK) yang telah diimplementasikan sejak tahun 2004. Informasi yang dihasilkan dari SIAMIK saat ini dapat dikatakan telah dapat digunakan untuk pengambilan keputusan tetapi sifatnya tidak fleksibel. Hal ini disebabkan karena informasi yang dihasilkan telah memiliki format yang tetap sehingga untuk penambahan atau pengurangan informasi perlu dilakukan perubahan pada SIAMIK.

Salah satu solusi yang dapat digunakan untuk memperoleh informasi adalah dengan
melakukan proses *query* secara langsung. Tetapi hal tersebut sulit untuk diimplementasikan karena adanya perbedaan bahasa antara pengguna informasi dan bahasa *Structure Query Language* (SQL) sebagai bahasa yang dikenal oleh *database*. Bagi pengguna yang mengerti SQL bukanlah menjadi suatu hambatan tetapi perlu mengetahui secara detail struktur *database*. Sedangkan bagi penggunan awam yang tidak mengerti SQL akan menjadi hambatan dalam memperoleh informasi.

Sehingga dapat diperoleh suatu permasalahan, "Bagaimana mengakomodasi perbedaan bahasa dari pengguna informasi dan komputer dalam memperoleh informasi akademik?". Berdasarkan permasalahan tersebut dapat diperoleh tujuan dari penelitian, yaitu dengan merancang *natural language processing* (NPL) sehingga pengguna informasi dapat melakukan memperoleh informasi akademik tanpa terkendala adanya perbedaan bahasa.

2. TINJAUAN PUSTAKA

2.1. Natural Language

Natural Language atau bahasa alami adalah bahasa yang dapat dimengerti dan dipahami oleh seseorang pada lokasi tertentu, sebagai contoh bahasa alami dari orang Indonesia adalah bahasa Indonesia. Bahasa alami biasa diajarkan orangtia pada anaknya ketika masih bayi. Seiring bertambahnya usia anak, maka pemahaman terhadap bahasa tersebut semakin meningkat. Untuk dapat mengetahui bahasa alami, perlu mengetahui susunan dari bahasa alami tersebut, sebagai contoh untuk Bahasa Inggris, seseorang perlu mempelajari kosa kata Inggris, tata bahasa Inggris, dan sebagainya.

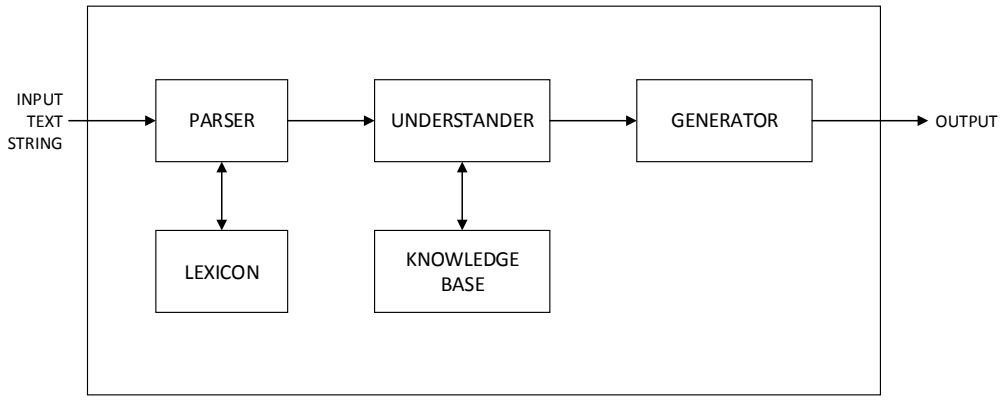
2.2. Natural Language Processing

Natural Language Processing (NLP) dapat didefinisikan sebagai kemampuan suatu komputer untuk memproses bahasa, baik lisan maupun tulisan yang digunakan oleh manusia dalam percakapan sehari-hari. Untuk proses komputasi, bahasa harus direpresentasikan sebagai rangkaian simbol yang memenuhi aturan tertentu. Secara sederhana, NLP adalah mencoba untuk membuat komputer dapat mengerti perintah-perintah yang ditulis dalam standar bahasa manusia.

Terdapat beberapa alasan yang menyulitkan NLP (Arman, 2004), yaitu masalah *ambiguity* atau makna ganda dan jumlah kosa kata (*vocabulary*) yang besar dan berkembang dari waktu ke waktu. Berdasarkan alasan tersebut, NLP tidak mempedulikan bagaimana suatu kalimat dimasukkan ke dalam komputer tetapi meng-*copy* informasi dari kalimat tersebut.

2.3. Sistem Pemahaman *Natural Language*

Pada Gambar 1, dapat dilihat 5 (lima) elemen utama dalam program NLP, yaitu *parser*, *lexicon*, *understander*, *knowledge base*, dan *generator*.



Gambar 1. Elemen Utama Sistem Pemahaman *Natural Language*

2.4. Parser

Elemen yang paling menentukan dalam NPL adalah *parser*. *Parser* merupakan bagian dari program yang menganalisa input secara sintaksistik. Setiap kata dan bagian-bagian ujarannya diidentifikasikan. *Parser* terdapat 3 (tiga) jenis, yaitu *Parser State-Machine*, *Parser Context-Free Recursive-Descent*, dan *Parser Noise-Disposal*. *Parser State Machine* menggunakan keadaan yang sesungguhnya dari kalimat untuk memprediksi tipe apa dari kata yang berlaku. *Parser Context-Free Recursive-Descent* menggunakan *production rule* untuk menganalisa sebuah kalimat.

Parser Noise-Disposal merupakan jenis *parser* yang sangat umum digunakan dalam aplikasi yang berbasis pada *database*. Contoh dari jenis *parser* ini misalkan terdapat query dalam bahasa alami sebagai berikut, "lihatkan saya semua perusahaan dengan persediaan > 100" maka

ke dalam bahasa SQL akan menjadi "select perusahaan from table_perusahaan where persediaan > 100".

2.5. Lexicon

Untuk menganalisis sintaksis, parser bekerja sama dengan *lexicon*. *Lexicon* berisi seluruh kata yang dikenali oleh program. *Lexicon* dapat berisi pula ejaan setiap kata yang benar dan merancang bagian ujarannya. Dalam implementasinya, *parser* merupakan alat untuk mencocokkan pola. Setelah salah satu kata diidentifikasikan, *parser* melacak *lexicon* untuk membandingkan setiap input kata dengan semua kata yang disimpan dalam *database*.

2.6. Understander dan Knowledge Base

Understander bekerja sama dengan basis pengetahuan untuk menentukan makna sebuah kalimat. Tugas *understander* adalah menyusun struktur data yang berisi makna, memahami kalimat input yang dimasukkan, dan menyimpannya ke dalam memori.

Dalam rangka menentukan apa yang dimaksudkan oleh input kalimat, sistem harus mengetahui kata-kata dan cara penyimpannya agar menjadi statemen bermakna. Tujuan dari *understander* adalah memanfaatkan output dari *parser* agar mengacu dari *knowledge base*. Apabila input kalimat berupa statemen, maka *understander* menentukan maknanya dengan melihat kata-kata atau frasa-frasa yang ada dalam basis pengetahuan.

2.7. Generator

Generator adalah input yang telah dimengerti untuk membuat *output* yang berguna. Struktur data yang dihasilkan oleh *understander* kemudian mengawali tindakan atau langkah berikutnya. Dalam bentuk paling sederhana, *generator natural language* memasukkan output jawaban standar yang telah dimasukkan terlebih dahulu kepada user didasarkan pada makna yang sudah diringkas dari input.

2.8. Database Language

Database Management Systems (DBMS) merupakan perantara bagi pengguna dengan *database* yang tersimpan *disk*. Cara berinteraksi atau berkomunikasi antara pemakai dengan *database* tersebut diatur dalam suatu bahasa khusus yang ditetapkan oleh perusahaan pembuat DBMS.

Bahasa khusus tersebut yang disebut dengan *database language* yang terdiri atas sejumlah perintah yang diberikan oleh pengguna dan dikenali oleh DBMS untuk memproses suatu aksi tertentu. *Database language* yang banyak dipakai adalah *Structured Query Language* (SQL).

2.9. Structured Query Language

SQL merupakan standar bahasa basis data relasional karena hamper semua DBMS telah mendukung penggunaan dari SQL.

SQL terbagi menjadi 2 (dua) bagian, yaitu *Data Definition Language* (DDL) dan *Data Manipulation Language* (DML).

DDL menyediakan perintah untuk mendefinisikan hubungan skema *database*, menghapus hubungan, dan memodifikasi skema relasi.

DML menyediakan kemampuan untuk permintaan informasi dari *database*, memasukkan *record*, menghapus *record*, dan memodifikasi isi *record*.

3. METODOLOGI PENELITIAN

3.1. Rancangan Penelitian

Rancangan penelitian yang akan dilakukan dalam melaksanakan proses penelitian ini adalah analisa kebutuhan, studi kelayakan, pendekatan terstruktur, dan uji kesahihan.

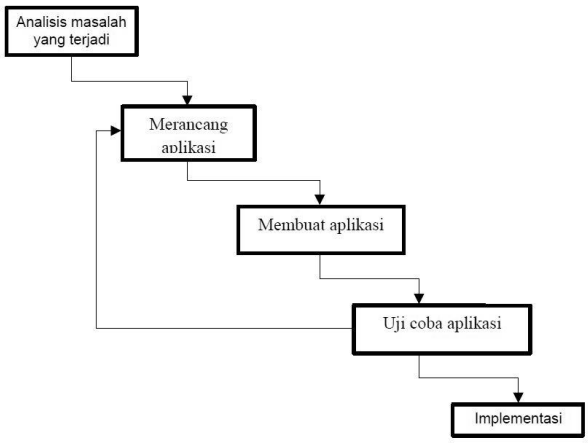
3.1.1. Analisa Kebutuhan

Analisa kebutuhan berperan dalam mengidentifikasi masalah atau kendala yang terdapat dalam proses ekstraksi informasi akademik di Universitas Atma Jaya Makassar. Tujuan dari analisa kebutuhan untuk mendapatkan fakta permasalahan yang ada dan sejalan dengan yang dikemukakan pada latar belakang dan

juga lebih memperkuat tujuan diadakannya penelitian dan perancangan.

3.1.2. Metode Perancangan

Metode perancangan yang akan digunakan adalah perancangan terstruktur dan langkah-langkah yang ditempuh dalam menyelesaikan perancangan ini dapat dilihat pada Gambar 2.



Gambar 2. Metode Perancangan Terstruktur

3.1.3. Uji Kesahihan

Tahap uji kesahihan ini dilakukan dengan melakukan pengujian pada aplikasi NPL untuk mengetahui apakah input yang dimasukkan telah memperoleh output yang sesuai dengan yang diharapkan. Tahap ini dilakukan untuk mengetahui tujuan awal yang merupakan target dari penelitian telah tercapai atau belum.

3.2. Ruang Lingkup Penelitian

Ruang lingkup dari penelitian ini adalah informasi akademik yang dikhususkan pada informasi akademik yang berhubungan dengan mahasiswa UAJM.

3.3. Metode Pengumpulan Data

Untuk menganalisis permasalahan dan kebutuhan yang akan dipenuhi oleh aplikasi yang akan dirancang, penulis mengumpulkan data dan informasi yang berasal dari berbagai sumber dengan menggunakan beberapa metode, seperti studi literatur dan observasi.

Studi literatur bertujuan untuk memperoleh informasi dari buku maupun dari internet yang berhubungan dengan permasalahan yang dihadapi. Studi litetarur menjadi landasan pengetahuan dalam penelitian ini.

Observasi bertujuan untuk melihat secara langsung kendala yang terjadi di lapangan, dalam hal ini pada lingkungan akademik UAJM. Dari kendala yang diperoleh tersebut akan dijadikan sebagai landasan untuk memperoleh solusi yang tepat dan sesuai. Selain itu, dilakukan pula analisa terhadap struktur *database* dari SIAMIK untuk mengetahui tabel dan atribut apa saja yang berhubungan dengan penelitian ini. Observasi juga dilakukan pada struktur SQL dan kemungkinan persamaan antara perintah SQL dan *natural language*.

3.4. Analisis Data

Metode analisa data yang dilakukan adalah analisa kualitatif. Analisa kuantitatif dilakukan dengan metode pengumpulan data melalui observasi. Dari hasil observasi, diharapkan dapat diperoleh permasalahan yang terjadi serta solusi yang tepat dari permasalahan yang dihadapi.

4. HASIL DAN PEMBAHASAN

4.1. Hasil Pengumpulan Data

Berdasarkan hasil observasi terhadap kebutuhan akan informasi akademik mahasiswa UAJM, dapat diperoleh bahwa informasi akademik mahasiswa dibutuhkan oleh beberapa pengambil keputusan, seperti Rektor, Wakil Rektor, Dekan, Wakil Dekan, dan Biro Akademik UAJM. SIAMIK yang ada saat ini secara umum telah mampu untuk memberikan informasi akademik yang dibutuhkan oleh para pengambil keputusan, tetapi terkadang masih membutuhkan informasi akademik yang belum dapat disediakan secara langsung oleh SIAMIK, seperti daftar mahasiswa yang memiliki Indeks Prestasi (IP) tertinggi dalam semester tertentu, daftar mahasiswa yang memiliki IP Kumulatif di bawah 2,75, dan sebagainya. Informasi yang khusus seperti ini tidak secara langsung dapat ditangani oleh SIAMIK tetapi pengolahan secara manual untuk memperoleh informasi tersebut. Untuk memenuhi kebutuhan informasi akademik tersebut secara langsung, maka perlu dilakukan *query* secara langsung untuk memperoleh informasi yang diinginkan. Proses *query* tersebut membutuhkan pemahaman terhadap SQL dan mengetahui struktur *database* dari SIAMIK yang dalam hal ini ditangani oleh Biro Administrasi Perencanaan dan Sistem Informasi (BAPSI).

Hasil observasi terhadap *database* SIAMIK UAJM dapat diperoleh bahwa untuk memperoleh informasi akademik yang berhubungan dengan mahasiswa, maka beberapa tabel yang berkaitan adalah tabel data_mahasiswa, data_krs, data_nilai_mahasiswa, dan beberapa tabel pendukung lainnya.

Penggunaan struktur *query* yang sederhana dalam memperoleh informasi dari *database* adalah SELECT atribut FROM tabel WHERE kondisi. Jika ditinjau dari *natural language*, maka penggunaan perintah SELECT dapat digantikan dengan kata “tampilkan”, “lihat”, dan kata lain yang memiliki arti kata yang sama. Sedangkan perintah WHERE dapat digantikan dengan kata “yang”, “dimana”, dan kata lainnya yang memiliki makna yang sama.

Berdasarkan hasil pengumpulan kata maka untuk mengatasi permasalahan yang dihadapi dalam memperoleh informasi akademik mahasiswa maka dapat digunakan NPL sebagai perantara antara *natural language* yang digunakan oleh pengambil keputusan dengan *database language* yang dikenali oleh DBMS sebagai penyimpan *database* SIAMIK UAJM.

4.2. Hasil Penelitian

NLP dapat didefinisikan sebagai kemampuan suatu komputer untuk memproses bahasa, baik lisan maupun tulisan yang digunakan oleh manusia dalam percakapan sehari-hari. Terdapat 5 (lima) elemen utama dalam program NLP, yaitu *parser*, *lexicon*, *understander*, *knowledge base*, dan *generator*.

4.2.1. *Parser*

Parser berfungsi untuk menganalisa setiap kata yang diinput dari pengguna dalam *natural language* kemudian memanfaatkan lexicon untuk mengecek apakah kata tersebut dikenali oleh *database language* atau tidak. Dalam penelitian ini, jenis parser yang digunakan *adalah Parser Noise-Disposal.*

Algoritma parser yang dirancang dalam penelitian ini adalah sebagai berikut:

1. *Scan* satu per satu kata yang berasal dari input teks yang dikirim oleh pengguna dalam *natural language*,
2. Lakukan pengecekan dalam *lexicon*, apakah kata yang di-*scan* terdapat dalam tabel relasi yang berisi seluruh terjemahan dari *natural language* ke dalam *database language*. Jika terdapat dalam *lexicon*, maka dilakukan pengecekan status dari arti masing-masing kata. Jika tidak maka lanjutkan *scan* ke kata berikutnya.

4.2.2. *Lexicon* dan *Knowledge Base*

Lexicon berfungsi sebagai penyimpan kata-kata yang dikenali oleh *database language*. Dalam penelitian ini,

lexicon disimpan dalam sebuah tabel yang memiliki struktur seperti pada Gambar 3.

#	Column	Type	Collation	Attributes	Null	Default
1	nama_natural	varchar(50)	utf8_general_ci		No	None
2	nama_sql	varchar(50)	utf8_general_ci		No	None
3	status	int(1)			No	None

Gambar 3. Struktur Tabel setara

Atribut nama_natural akan menyimpan daftar keseluruhan dari penggunaan *natural language* yang memiliki penyetaraan dengan nama_sql atau *database language* yang telah diinput sebelumnya.

Knowledge base bertugas untuk membantu *understander* dengan menyimpan makna dari arti kalimat yang telah melalui proses *parser*. Atribut status merupakan *flag* (penanda) dari arti kata dalam *database language*, yang memiliki kemungkinan nilai 0 (nol) sebagai sintaks, 1 (satu) sebagai atribut, dan 2 (dua) sebagai tabel.

Dalam penelitian ini, *knowledge base* juga bertugas untuk menyimpan hubungan relasi antar tabel yang berhubungan. Hubungan relasi antar tersebut disimpan dalam sebuah tabel yang memiliki struktur seperti pada Gambar 4.

#	Column	Type	Collation	Attributes	Null	Default
1	tabel	varchar(50)	utf8_general_ci		No	None
2	tabel1	varchar(50)	utf8_general_ci		No	None
3	kunci	varchar(50)	utf8_general_ci		No	None

Gambar 4. Struktur Tabel relasi

Atribut tabel dan tabel1 menyimpan nama tabel yang saling berelasi dengan kunci yang menghubungkan tabel tersebut disimpan dalam atribut kunci.

4.2.3. *Understander*

Tujuan dari *understander* adalah memanfaatkan output dari *parser* agar mengacu dari *knowledge base*. Apabila input kalimat berupa statemen, maka *understander* menentukan maknanya dengan melihat kata-kata atau frasa-frasa yang ada dalam basis pengetahuan.

Algoritma *understander* yang dirancang dalam penelitian ini adalah sebagai berikut:

1. Melakukan pengecekan status dari *output* yang dihasilkan oleh *parser*,
2. Pengecekan status dilakukan untuk mengetahui letak atau posisi dari arti kata tersebut. Setiap arti kata memiliki status masing-masing yang terdiri dari 3 (tiga) kemungkinan, yaitu 0, 1, dan 2. Apabila status 0 (nol) berarti merupakan suatu sintaks, jika status 1 (satu) berarti merupakan suatu atribut, dan status 2 (dua) sebagai penanda tabel,
3. Apabila ditemukan arti kata yang berupa sintaks "where" dan ditemukan arti kata berikutnya adalah atribut, maka selanjutnya pasti berupa operator dan nilai,
4. Apabila ditemukan arti kata yang berupa sintaks "and", "or", atau "not" maka kembali ke langkah nomor 3 (tiga),
5. Apabila ditemukan arti kata yang berupa sintaks "order by" atau "group by" maka arti kata yang berikutnya berupa atribut dan dapat diikuti dengan operator pengurutan,
6. Simpan setiap arti kata ke dalam sebuah *array* sesuai dengan statusnya masing-masing, apakah status sintaks, atribut, tabel, kondisi, atau pengurutan,
7. Lakukan pengecekan relasi antar tabel dengan membandingkan antar isi *array* yang menyimpan nama tabel yang digunakan. Apabila memiliki relasi, maka tambahkan isi *array* yang menyimpan kondisi dengan statemen yang menandakan adanya relasi antar tabel.

4.2.4. *Generator*

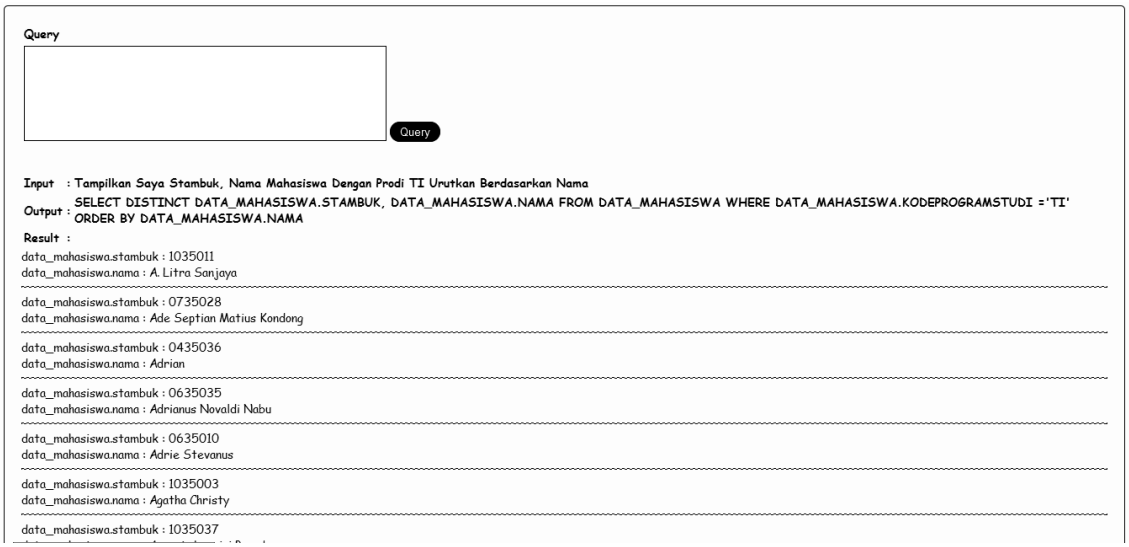
Generator adalah input yang telah dimengerti untuk membuat *output* yang berguna atau dalam penelitian ini, *output* yang akan dihasilkan oleh *generator* adalah SQL yang diperoleh berdasarkan hasil proses *parser* dan *understander*. *Generator* untuk menghasilkan SQL adalah dengan berlandaskan pada struktur perintah SQL dan membuat pemanggilan secara berurut

dari isi *array* mulai dari statemen, atribut, tabel, kondisi, dan pengurutan.

SQL yang telah dihasilkan kemudian dieksekusi ke dalam *database* SIAMIK UAJM untuk memperoleh informasi akademik mahasiswa sesuai dengan yang diinginkan.

4.2.5. Hasil Perancangan Aplikasi NPL

Aplikasi NPL yang dirancang menggunakan bahasa pemrograman PHP dan *database* MySQL. Input yang diharapkan dari aplikasi adalah berupa *natural language* dari pengguna dan akan memperoleh *output* berupa hasil pemrosesan dalam perintah SQL dan informasi akademik yang diharapkan.

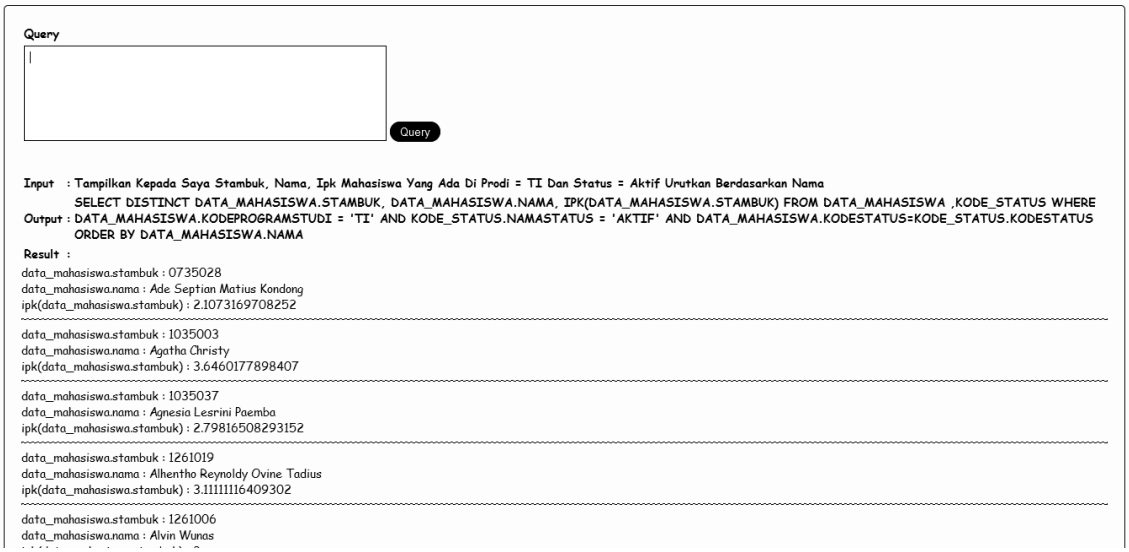


Gambar 5. Tampilan Output dari Aplikasi NPL

4.3. Uji Kesahihan

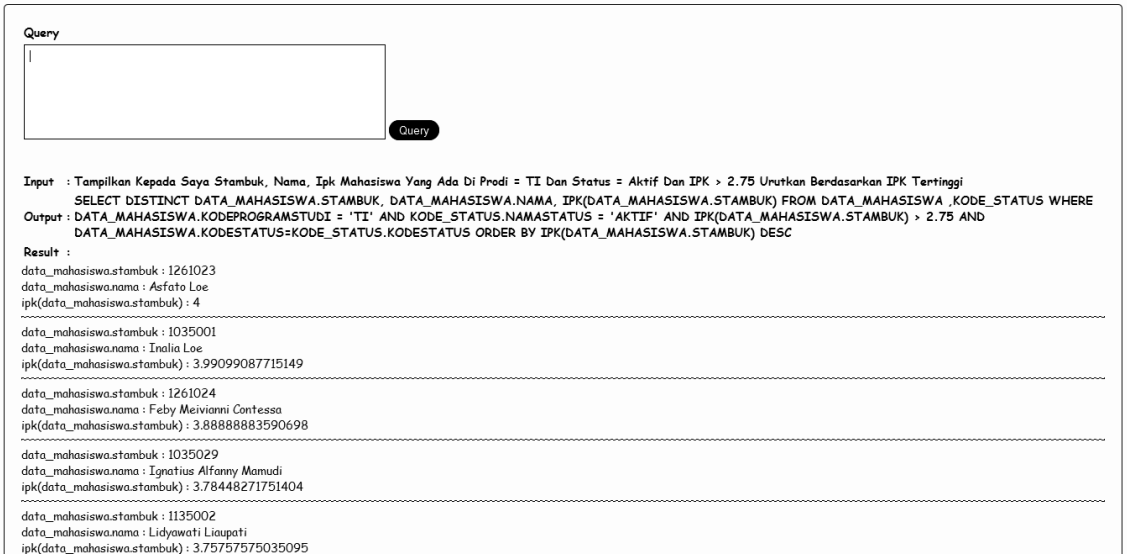
Uji kesahihan dilakukan untuk melakukan pengecekan terhadap input yang dimasukkan dalam *natural language* pengguna, kemudian melakukan pengecekan apakah output yang dihasilkan telah sesuai dengan yang diharapkan atau tidak.

Uji kesahihan yang pertama adalah dengan mencoba memberikan input untuk menampilkan daftar mahasiswa yang terdaftar di program studi Teknik Informatika dan berstatus aktif.



Gambar 6. Tampilan Hasil Informasi dari Mahasiswa TI dan Status Aktif

Uji kesahihan yang kedua adalah dengan mencoba memberikan input untuk menampilkan daftar mahasiswa yang terdaftar di program studi Teknik Informatika dan berstatus aktif dan memiliki IPK lebih besar dari 2,75 dan diurutkan berdasarkan IPK tertinggi.



Gambar 7. Tampilan Hasil Informasi IPK Mahasiswa

Berdasarkan hasil output dari kedua input (Gambar 6 dan Gambar 7) dapat dikatakan bahwa output yang dihasilkan telah sesuai dengan input yang dimasukkan oleh pengguna.

5. KESIMPULAN

Kesimpulan yang dapat diperoleh dari penelitian ini adalah sebagai berikut:

- NPL dapat digunakan untuk membantu pengguna awam dalam memperoleh informasi tanpa perlu mempelajari *database language* dan struktur *database* SIAMIK UAJM.
- NPL sangat bergantung pada *lexicon* dan *knowledge base* yang dimilikinya, sehingga diperlukan adanya pendataan kosakata yang digunakan oleh pengguna sebelum menggunakan aplikasi NPL.

6. DAFTAR PUSTAKA

[1] Desiani, Anita, dan Arhani, Muhammad. 2006. *Konsep Kecerdasan Buatan*. Palembang: Penerbit Andi.

[2] Fathansyah. 1999. *Basis Data*. Bandung: Penerbit Informatika.

[3] Jogiyanto. 2008. *Analisis & Desain Sistem Informasi: Pendekatan Terstruktur Teori Dan Praktek Aplikasi Bisnis*. Yogyakarta: Penerbit Andi.

[4] Silberschatz, Abraham., Korth, Henry F., Sudarshan, S. 2009. *Database System Concepts Sixth Edition*. New York: McGraw Hill.

[5] Siswanto. 2010. *Kecerdasan Tiruan Edisi 2*. Jakarta: Graha Ilmu.

[6] Suparman dan Marlan. 2005. *Komputer Masa Depan: Pengenalan Artificial Intelligence*. Bandung: Penerbit Andi.

[7] Suyoto. 2004. *Intelegensi Buatan: Teori dan Pemrograman*. Yogyakarta: Penerbit Gava Media.