PERANCANGAN TRANSLATOR BAHASA ALAMI KE DALAM FORMAT SQL (*STRUCTURED QUERY LANGUAGE*) DENGAN FOKUS PADA KASUS RETRIVAL INFORMASI

Ihsan Faturohman[1], Galih Hermawan[2]

Teknik Informatika – Universitas Komputer Indonesia
Jl. Dipatiukur No. 112-116, Bandung 40132
E-mail : ihsan.seeking@gmail.com[1], galih.hermawan@email.unikom.ac.id[2]

ABSTRAK

Database merupakan tempat penyimpanan suatu informasi, namun tidak semua orang dapat mengakses informasi tersebut, karena memerlukan suatu keahlian khusus, yaitu mengerti format baku yang disebut *syntax* SQL (*Structured Query Language*). Tidak semua perintah masukan yang digunakan untuk mengakses informasi berupa *syntax* SQL. Seringkali perintah masukan berupa bahasa alami. *Syntax* SQL tentu memiliki format yang berbeda dengan bahasa alami indonesia. Oleh karena itu diperlukan suatu alat bantu yang dapat menerjemahkan bahasa alami ke dalam format SQL sebagai jembatan untuk mengakses informasi dari DBMS. Sistem Penerjemah bahasa alami yang dibangun menggunakan metode *rule-base* untuk menentukan tahapan penerjemahan, berdasarkan hasil ekstraksi informasi dari kalimat perintah. ada empat blok proses, yaitu analisis database untuk untuk mengekstrak informasi database, lalu *preprocesing* kalimat untuk mengidealkan kalimat masukan, setelah itu masuk tahap Proses penerjemahan, yang terdiri dari identifikasi perintah, tabel, atribut, fitur, dan penyusunan *query*. terahir hasil *query* diekseskusi untuk diuji keberhasilannya dalam menampilkan data yang dimaksud. Berdasarkan hasil pengujian fungsi dari antarmuka sistem telah berjalan cukup baik. *output* dari sistem ini memiliki akurasi berkisar antara 72.22% - 83.87% untuk pengujian 6 tipe *query* yang berjumlah total 18 kalimat pengujian.

Kata kunci : Ekstraksi Informasi, *Rule - Based*, Query, NLP, Bahasa Alami.

1. PENDAHULUAN

Database merupakan tempat penyimpanan suatu informasi, namun tidak semua orang dapat mengakses informasi tersebut, karena memerlukan suatu keahlian khusus, yaitu mengerti format baku yang disebut *syntax* SQL (*Structured Query Language*). *Syntax* SQL adalah suatu bahasa yang digunakan oleh DBMS MySQL. DBMS (*Database Management System*) sendiri adalah sebuah *engine* atau *software* yang digunakan untuk mengelola *database*. Pengolahan struktur dan data pada DBMS juga dibagi 2 yaitu DDL (*Data Definition Language*) dan DML (*Data Manipulation Language*). DDL adalah struktur *syntax* SQL yang berfokus pada pembangunan sebuah wadah untuk informasi sedangkan DML adalah struktur *syntax* SQL yang berfokus pada manipulasi data, seperti, menyimpan, mengubah, menghapus, dan menampilkan suatu informasi. Tidak semua perintah masukan yang digunakan untuk mengakses informasi berupa *syntax* SQL. Seringkali perintah masukan berupa bahasa alami. *Syntax* SQL tentu memiliki format yang berbeda dengan bahasa alami indonesia. Oleh karena itu diperlukan suatu alat bantu yang dapat menerjemahkan bahasa alami ke dalam format SQL sebagai jembatan untuk mengakses informasi dari DBMS.

Peneleitian ini fokus terhadap ekstraksi informasi dalam penerjemahan kalimat ke dalam syntax SQL untuk mengakses informasi yang ada pada DBMS. karena penelitian ini berfokus pada kasus *retrival* informasi maka beberapa batasan masalah diterapkan sebagai berikut:

a. *Syntax* SQL yang dianalisis pada penelitian ini berfokus pada *Retrival* Informasi yaitu *query SELECT*.
b. *Query SELECT* yang dilakukan memiliki 6 tipe, yaitu, *query* tanpa kondisi, *query* dengan kondisi, *query* Banyak kondisi, *query* banyak tabel tanpa kondisi, *query* banyak tabel banyak kondisi, *query order*.

Adapun tujuan yang ingin dicapai dari penelitian ini adalah:

a. Menyediakan sistem yang berfungsi sebagai penerjemah bahasa alami ke dalam format SQL sebagai solusi untuk mengakses informasi suatu *database*.
b. Menghitung akurasi sistem yang akan dibangun untuk mengetahui apakah sistem dapat menerjemahkan perintah ke dalam format SQL dan menampilkan data yang diminta.

2. TINJAUAN PUSTAKA

2.1 Penelitian Sebelumnya

Alat bantu yang dibutuhkan adalah *sub*-sistem yang dapat memberikan jawaban atas pertanyaan yang diajukan oleh pengguna dengan memasukan

pertanyaan dalam bahasa alami Indonesia yang dipahami oleh pengguna. Ada beberapa penelitian mengenai alat bantu ini seperti penelitian yang dilakukan oleh Setyawan Wibisono dalam penelitian bahasa alami untuk basis data akademik dengan format data XML dengan menambahkan pola-pola kalimat yang dapat menjadi referensi pengguna aplikasi tersebut [1]. Penelitian yang dilakukan oleh Nendi Isharmawan dalam penelitian tentang penerjemah bahasa alami menjadi bahasa SQL dengan batasan database yang digunakan untuk dianalisis hanya satu database, dan kasus yang dicontohkan mengakses 1 tabel [2]. Penelitian yang dilakukan oleh Muhammad Habibi Taqwim dalam melakukan penelitian untuk membangun sebuah sistem yang dapat menganalisis soal cerita dan jawabannya dengan menerapkan metode-metode pemrosesan bahasa alami khususnya ekstraksi informasi agar dapat menghasilkan langkah-langkah penyelesaian soal cerita untuk mempermudah siswa mengecek kembali jawaban latihan soal yang ada, di mana dalam penelitian ini metode yang digunakan dapat diterapkan dalam kasus penerjemahan ini, karena memiliki karakteristik yang sama yaitu menjawab pertanyaan dalam bahasa Indonesia [3].

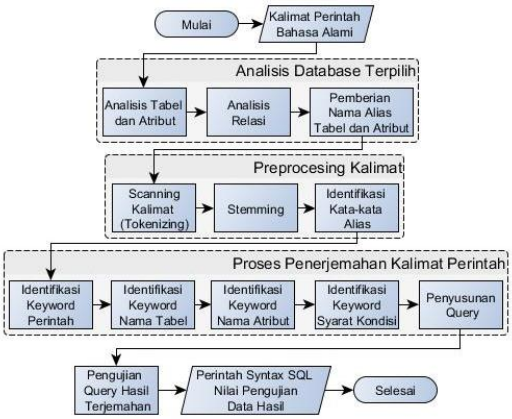
Berdasarkan uraian di atas, perlu diadakan penelitian untuk membangun sebuah sistem yang dapat menganalisis pertanyaan, dengan menerapkan metode – metode pemrosesan bahasa alami, khususnya ekstraksi informasi agar dapat menghasilkan langkah – langkah untuk menerjemahkan pertanyaan ke dalam format SQL yang lebih dinamis dengan kasus banyak tabel.

2.2 Landasan Teori

Pada penelitian ini berkaitan dengan banyak teori diantaranya ada, *Natural Language Processing* (NLP) yaitu bidang ilmu komputer dan linguistik yang berkaitan dengan interaksi antara manusia dengan komputer[4], *Rule-based System* yaitu jalan untuk menyimpan atau memanipulasi pengetahuan untuk menintrepentasikan informasi yang berguna[5], Ekstraksi Informasi yaitu Untuk mendapatkan informasi terstruktur dari teks yang tidak terstruktur, hal pertama yang perlu didefinisikan adalah informasi target sebagai informasi terstruktur yang akan diekstrak[6].

3. METODE TRANSLASI

Metode dalam menerjemahkan perintah dalam bahasa alami ke dalam *syntax* SQL ini memliki beberapa tahap diantaranya, Menganalisis database untuk mendapatkan info tabel, atribut dan relasi antar tabel, lalu menambahkan kata alias di setiap tabel dan atribut untuk memperkaya *keyword* dari nama tabel dan atributnya. Selanjutnya tahap proses penerjemahan, sebelum ditejemahkan kalimat akan melalui proses *Preprocessing* yaitu, tahap *tokenizing* untuk memecah kalimat, lalu tahap *stemming* untuk menghilangkan kata berimbuhan, dan tahap alias untuk menganti kata alias yang digunakan di kalimat menjadi kata yang dimaksud. Setelah itu baru dilakukan proses penerjemahan yang dibagi kedalam 5 proses yaitu, proses identifikasi perintah, identifikasi tabel dan relasi, identifikasi atribut, identifikasi fitur, dan proses penyusunan *Query*. Terakhir dilakukan proses pengujian untuk menampilkan hasil eksekusi *Syntax* SQL berupa tabel hasil eksekusi. Blok proses umum sistem dapat dilihat di Gambar 1.



Gambar 1. Blok Proses Umum Sistem

3.1 Kalimat Perintah Masukan

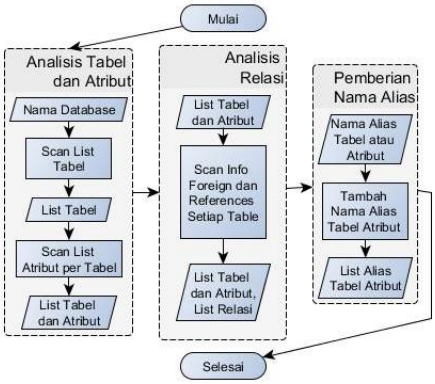
Pada proses ekstraksi informasi, data masukan berbentuk kalimat perintah bahasa Indonesia yang dimasukkan oleh pengguna. Contoh kalimat perintah masukan berikut

“ Tampilkan judul film yang termasuk kategori comedy, dimana dalam deskripsi film tersebut mengandung kata "dentist" ”

Untuk Contoh Kasus Penerjemahan Kalimat Perintah tersebut *database* yang dipilih adalah *database* sakila[7] yang memiliki 23 tabel dan berelasi.

3.2 Analisis *Database*

Tahapan paling awal sebelum memasukan kalimat perintah adalah memilih *database* yang akan diakses dan mendaftarkan kata alias untuk mempermudah dalam penulisan kalimat perintah. Blok proses analisis *database* apa dilihat di Gambar 2.



Gambar 2. Blok Proses Analisis *Database*

Pada blok ini sistem menganalisis *database* yang dipilih untuk digunakan. Ada 3 proses dalam blok ini yaitu, Analisis Tabel dan Atribut, Analisis Relasi, dan Pemberian Nama Alias. Analisis Tabel dan Atribut adalah proses mengidentifikasi semua nama tabel dan atribut yang ada pada *database* tersebut. Analisis Relasi adalah proses menelusuri tabel mana saja yang berelasi. Pemberian Nama Alias adalah proses di mana pengguna dapat mendaftarkan kata alias terhadap nama tabel atau atribut yang telah di identifikasi sebelumnya.

a. Analisis *Database* Tabel Atribut

Ketika *database* dipilih maka sistem akan memindai semua tabel yang terdapat dalam *database* tersebut, lalu dari setiap tabel yang dipindai juga akan dipindai semua atributnya, sehingga didapatkanlah daftar tabel beserta atributnya. Berikut Tabel 1 sebagian Hasil Analisis Nama Table dan atributnya.

Tabel 1. Hasil Analisis Nama tabel dan Atribut

#Tbl	#Attr	Nama Tabel / Atribut	Tipe
1	**actor**		Base Tabel
	1	actor_id	smallint(5) unsigned
	2	first_name	varchar(45)
	3	last_name	varchar(45)
	4	last_update	timestamp
...	...	...	...
23	4	last_update	timestamp

b. Analisis Relasi Tabel

Setelah nama tabel dan atributnya dipindai, selanjutnya mencari relasi antar tabel yang telah dipindai dengan cara memindai *keyword constraint*, *foreign key*, dan *references* di *information schema* menggunakan perintah "*show create table* <nama tabel>" di setiap tabel yang dipindai. Tabel 2 adalah contoh hasil analis relasi.

Tabel 2. Hasil Analisis Relasi Tabel

#	Nama Relasi	*Foreign Key* (nama_table. nama_attribute)	*References* (nama_table. nama_attribute)
1	fk_address_city	address.**city_id**	city.**city_id**
...	...	...	...
22	fk_store_staff	store.**manager_staff_id**	store.**manager_staff_id**

c. Pemberian Nama Alias

Pemberian nama alias pada tabel atau atribut dilakukan oleh pengguna dan bersifat *optional*. Tabel 3 adalah contoh pemberian nama alias.

Tabel 3. Hasil Pemberian Nama Alias

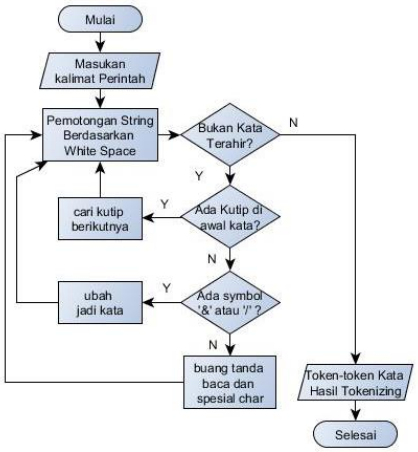
#Tbl	#Attr	Nama Tabel / Atribut	Alias
1	**actor**		"aktor" "pemain"
	1	actor_id	
	2	first_name	"nama" "aktor"
	3	last_name	"lenkap"
	4	last_update	
...	...	...	...
23	4	last_update	

3.3 *Preprocessing* Kalimat

Tahapan *preprocessing* ini bertujuan untuk mempermudah penerapan metode *rule based* dalam penyelesaian soal cerita. Pada Blok kedua *Preprocessing* kalimat ada 3 proses yaitu, *Tokenizing*, *Stemming* dan Identifikasi kata alias. *Tokenizing* adalah memecah kalimat menjadi *token-token* kata. *Stemming* adalah proses pencarian kata dasar dari setiap *token* hasil *tokenizing*. Proses Identifikasi kata alias adalah mencari kata-kata alias yang terdaftar untuk diganti oleh nama tabel atau atribut yang dimaksud.

a. *Tokenizing*

Pada tahap *tokenizing* ini dilakukan pemotongan *string* berdasarkan setiap kata penyusunnya. Berikut Gambar 3 adalah alur *preprocessing tokenizing*.



Gambar 3. Alur Proses *Tokenizing*

Pemotongan tersebut dilakukan ketika masukan memiliki *white space*. Karakter khusus selain huruf dan angka akan dianggap sebagai kata penyusun *string* apabila karakter tersebut berdiri sendiri dan dipisahkan oleh spasi. Karakter tersebut dapat dilihat pada lampiran. Tabel 4 adalah contoh hasil proses tahap *tokenizing* pada kalimat perintah.

Tabel 4. Hasil Proses *Tokenizing*

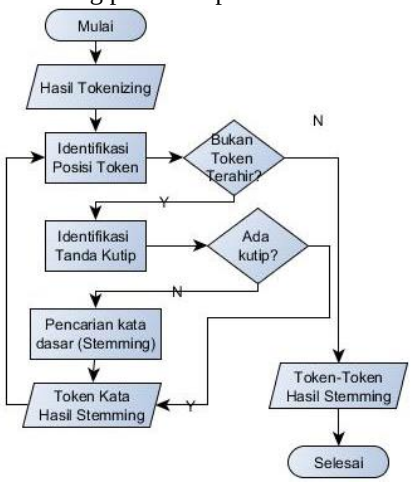
Nama	Objek	
Input Kalimat	Tampilkan judul film yang termasuk kategori comedy, dimana dalam deskripsi film tersebut mengandung kata "dentist"	
Output	Token 1 : Tampilkan Token 2 : judul	Token 9 : dalam Token 10 : deskripsi Token 11 : film

Nama	Objek	
	Token 3 : film Token 4 : yang Token 5 : termasuk Token 6 : kategori Token 7 : comedy Token 8 : dimana	Token 12 : tersebut Token 13 : mengandung Token 14 : kata Token 15 : 'dentist'

Pada Tabel 4 di atas dapat diamati bahwa setiap kata dan karakter penyusun teks input yang dipisahkan oleh spasi (white space) akan dikategorikan sebagai token oleh sistem. Setiap kata yang mengunakan kutip akan dijadikan setiap kata Siantar kutip tersebut jadi sebuah token, dan kutipnya menjadi kutip satu. Untuk simbol seperti '&' atau '/' akan di ubah jadi kata yang dimaksud. Lalu untuk setiap tanda baca akan dihilangkan.

b. Stemming

Stemming adalah proses mengubah kata berimbuhan menjadi kata dasar. Aturan-aturan bahasa diterapkan untuk menanggalkan imbuhan-imbuhan itu. Berikut adalah alur dari penerapan proses stemming pada setiap token hasil tokenizing.



Gambar 4. Alur Proses Stemming

Proses stemming mengunakan Sastrawi stemmer yang merupakan sebuah library stemmer sederhana yang didesain untuk dapat digunakan secara mudah. Proses stemming dilakukan ke setiap token hasil tokenizing. Hasil dari stemming ini juga menghilangkan karakter-karakter khusus yang menempel pada setiap token. Tabel 5 adalah contoh hasil proses tahap stemming pada token-token.

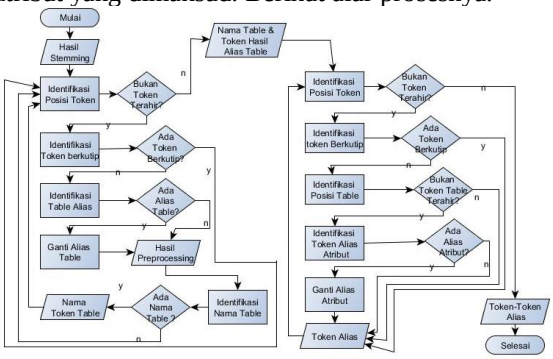
Tabel 5. Hasil Proses Stemming

Nama	Objek	
Output	Token 1 : Tampilkan => tampil Token 2 : judul Token 3 : film Token 4 : yang Token 5 : termasuk Token 6 : kategori Token 7 : comedy Token 8 : dimana	Token 9 : dalam Token 10 : deskripsi Token 11 : film Token 12 : tersebut => sebut Token 13 : mengandung => kandung Token 14 : kata Token 15 : 'dentist'

Pada Tabel 5 di atas dapat diamati bahwa setiap token yang berimbuhan akan diganti dengan kata dasarnya, kecuali kata tersebut salah satu dari daftar kata yang tidak perlu distemming atau token tersebut menggunakan kutip. Token yang melewati proses stemming, setiap hurufnya menjadi lowercase.

c. Identifikasi Alias

Pada tahap ini dilakukan identifikasi kata-kata alias, untuk mengganti kata-kata tersebut menjadi makna sebenarnya yaitu nama tabel atau nama atribut yang dimaksud. Berikut alur prosesnya.



Gambar 5. Alur Identifikasi Alias

Pada proses ini dilakukan dua kali perulangan, yaitu perulangan pertama untuk identifikasi kata alias tabel dan diganti dengan nama tabel yang dimaksud sekaligus identifikasi nama-nama tabel yang teridentifikasi dalam token-token tersebut, lalu perulangan kedua untuk mencari kata alias atribut dari setiap token berdasarkan daftar alias atribut dari tabel yang teridentifikasi sebelumnya, dan diganti juga dengan nama atribut yang dimaksud. Berdasarkan kata alias yang sebelumnya dicontohkan maka hasil setelah proses identifikasi kata alias dapat di lihat di Tabel 6.

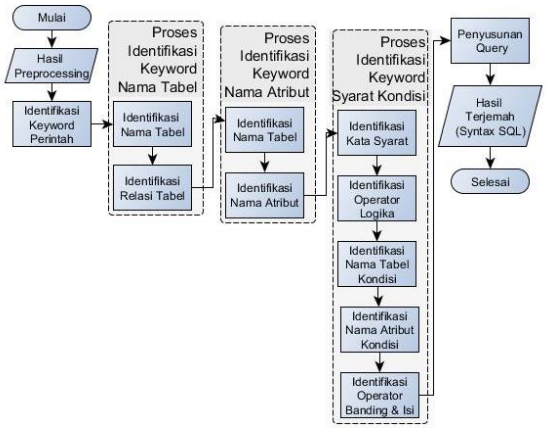
Tabel 6. Hasil Proses Identifikasi Alias

Nama	Objek	
Input	Token 1 : tampil Token 2 : judul => title Token 3 : film Token 4 : yang Token 5 : termasuk Token 6 : kategori => category Token 7 : comedy Token 8 : dimana	Token 9 : dalam Token 10 : deskripsi => description Token 11 : film Token 12 : sebut Token 13 : kandung Token 14 : kata Token 15 : 'dentist' => dentist

Pada Tabel 6 di atas dapat kita ketahui ada beberapa kata alias yaitu, token "judul" menjadi "title", token "kategori" menjadi "category", token "deskripsi" menjadi "description", dan untuk toke yang berkutip, kutipnya dihilangkan.

3.4 Proses Penerjemahan Kalimat

Hasil dari tahap preprocessing kalimat berupa list token yang siap diolah pada tahap penerjemahan kalimat perintah. Berikut Gambar 6 adalah blok Proses Penerjemahan kalimat perintah.

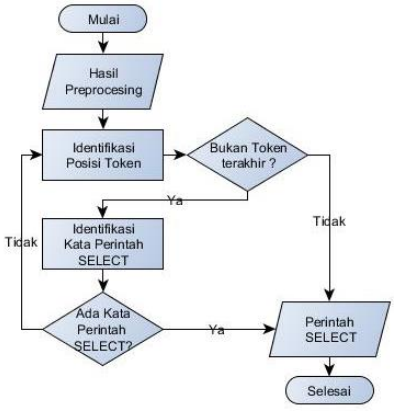


Gambar 6. Blok Proses Penerjemah Kalimat

Tahap ini menerapkan rule based system demi mendapatkan informasi dari hasil preprocessing dan mengolahnya sedemikian rupa sehingga dapat menerjemahkan kalimat perintah tersebut. Pada Blok proses ini ada 5 sub proses yaitu. Identifikasi keyword perintah, Identifikasi nama tabel, identifikasi nama atribut, Identifikasi syarat kondisi, dan penyusunan Query. Identifikasi perintah adalah proses mencari kata perintah dalam tokent-token. Identifikasi nama tabel adalah proses mencari nama tabel dari token-token. Identifikasi nama atribut adalah proses mencari atribut apa saja yang akan ditampilkan. Identifikasi syarat kondisi adalah proses mencari apakah ada tanda-tanda soal menggunakan fasilitas Where kondisi. Penyusunan query adalah proses yang mengumpulkan semua hasil identifikasi dan menyusunnya dalam Syntax SQL.

a. Identifikasi Perintah

kata-kata yang mengindikasikan penggunaan SELECT dicari dalam list hasil preprocesing pada proses ini. Berikut Gambar 7 adalah alur proses identifikasi keyword Perintah.



Gambar 7. Alur Proses Identifikasi Perintah

Identifikasi kata perintah dilakukan pada setiap token hasil preprocesing. Jika teridentifikasi maka variabel perintah akan ditentukan sesuai kelompok kata perintah sebagai berikut :

Tabel 7. Kata Perintah

No	Perintah	Daftar Kata
1	SELECT	tampil, apa, berapa, siapa, cari, lihat

Berikut ini merupakan contoh analisis identifikasi perintah pada token-token hasil Preprocessing.

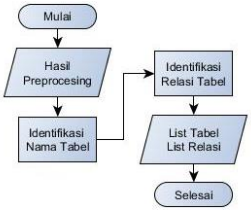
Tabel 8. Hasil Proses Identifikasi Kata Perintah

Nama Objek	Token Objek	
[Masukan] Array Token Preprocessing	Token 1 : tampil => SELECT Token 2 : title Token 3 : film Token 4 : yang Token 5 : termasuk Token 6 : category Token 7 : comedy Token 8 : dimana	Token 9 : dalam Token 10 : description Token 11 : film Token 12 : sebut Token 13 : kandung Token 14 : kata Token 15 : dentist
[Keluaran] Identifikasi perintah	Ident_perintah : SELECT	

Pada Tabel 8 kata perintah “tampil” ditemukan di token 1, jadi perintah yang teridentifikasi adalah “SELECT”.

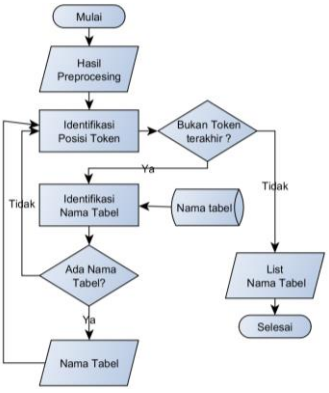
b. Identifikasi Table

Kalimat perintah dalam kasus ini selalu berkaitan dengan tabel-tabel dalam database, untuk mengetahui tabel apa saja yang ada dalam kalimat, maka dilakukan pencarian keyword nama tabel dari token-token hasil preprocesing, setiap tabel memiliki kemungkinan menggunakan relasi, maka perlu dicari relasi tabel tersebut.



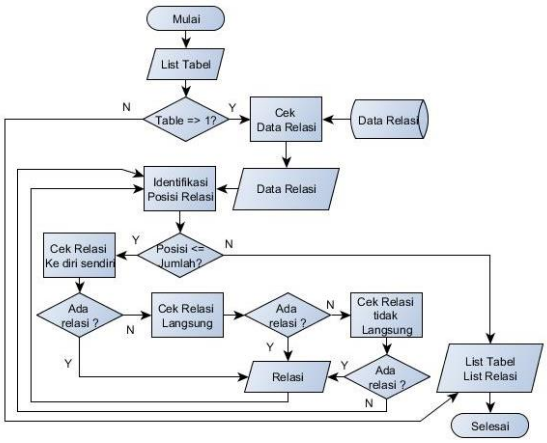
Gambar 8. Alur Proses Identifikasi Tabel dan Relasi

Dalam Tahap ini dilakukan dua proses, yaitu: proses identifikasi nama tabel dan proses identifikasi relasi tabel. Token kata yang mengindikasikan nama tabel dicari dalam proses identifikasi nama tabel seperti Gambar 9.



Gambar 9. Alur Proses Identifikasi Nama Tabel

Proses identifikasi nama tabel ini dilakukan ke setiap token yang ada dengan membandingkan dengan nama tabel yang ada pada database pendukung. Setiap tabel hasil identifikasi nama tabel akan di cek relasinya seperti pada Gambar 10.



Gambar 10. Alur Proses Identifikasi Relasi

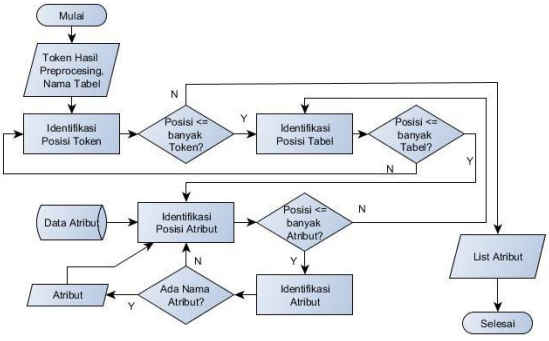
Pada proses ini dilakukan beberapa kali pengecekan diantaranya cek relasi ke diri sendiri, cek relasi ke langsung, dan cek relasi tidak langsung.

Tabel 9. Hasil Proses Identifikasi Tabel dan Relasi

Nama Objek	**Token Objek**	
[Masukan] Array Token Preprocessing	Token 1 : tampil Token 2 : title Token 3 : film => film Token 4 : yang Token 5 : termasuk Token 6 : category => category Token 7 : comedy Token 8 : dimana	Token 9 : dalam Token 10 : description Token 11 : film => film Token 12 : sebut Token 13 : kandung Token 14 : kata Token 15 : dentist
[Keluaran] Identifikasi Tabel	ident_table[1] : film ident_table[2] : category	
[Keluaran] Identifikasi Relasi	ident_relasi[1] : FROM `film` ident_relasi[2] : INNER JOIN `film_category` ON `film`.`film_id` = `film_category`.`film_id` ident_relasi[3] : INNER JOIN `category` ON `film_category`.`category_id` = `category`.`category_id`	

c. Identifikasi Atribut

Token kata yang mengindikasikan nama atribut dari tabel yang teridentifikasi dicari dalam proses ini.



Gambar 11. Alur Proses Identifikasi Atribut

Pada proses ini pengecekan nama atribut dilakukan ke setiap token dengan dibandingkan ke setiap atribut milik tabel yang yang teridentifikasi.

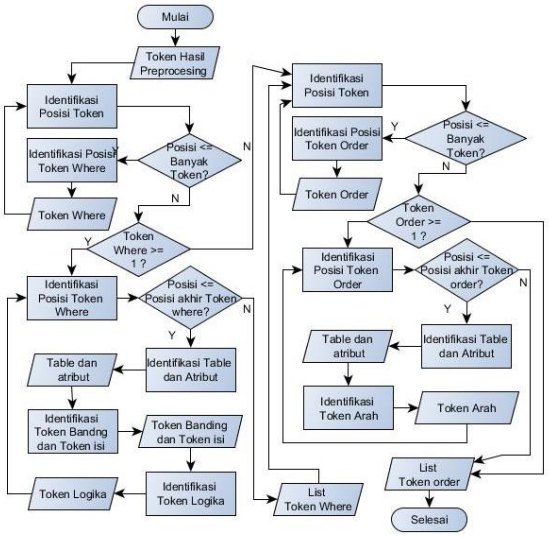
Tabel 10. Hasil Proses Identifikasi Atribut

Nama Objek	**Token Objek**	
[Masukan] Identifikasi Tabel	ident_table[1] : film ident_table[2] : category	
[Masukan] Array Token Preprocessing	Token 1 : tampil Token 2 : title => `title` Token 3 : film Token 4 : yang Token 5 : termasuk Token 6 : category Token 7 : comedy Token 8 : dimana	Token 9 : dalam Token 10 : description => `description` Token 11 : film Token 12 : sebut Token 13 : kandung Token 14 : kata Token 15 : dentist
[Keluaran] Identifikasi Atribut	ident_attribute[1] : `film`.`title` AS 'judul' ident_attribute[2] : `film`.`description` AS 'deskripsi'	

Pada Tabel 10 ditemukan dua atribut dari tabel film.

d. Identifikasi Fitur

Pada Tahap proses ini akan dicari token kata yang mengindikasikan fitur where atau fitur order.



Gambar 12. Alur Proses Identifikasi Fitur

Pada proses ini dilakukan pengecekan fitur WHERE dan ORDER, setiap fitur melakukan pengecekan sesuai kebutuhan fiturnya.

Tabel 11. Hasil Proses Identifikasi Fitur

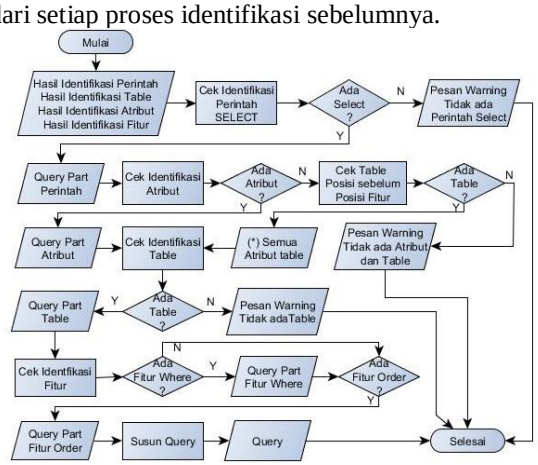
Nama Objek	**Token Objek**	
[Masukan] Array Token Preprocessing	Token 1 : tampil Token 2 : title Token 3 : film Token 4 : yang => Fitur Kondisi Token 5 : termasuk	Token 7 : comedy Token 8 : dimana => Fitur Kondisi Token 9 : dalam Token 10 : description Token 11 : film Token 12 : sebut

Nama Objek	Token Objek	
	Token 6 : category	Token 13 : kandung Token 14 : kata Token 15 : dentist
[Keluaran] Identifikasi Kondisi	ident_where[1] : yang ident_where[2] : dimana	
[Keluaran] Identifikasi Operator Logika	ident_where_logika[1] : AND	
[Keluaran] Identifikasi Tabel Kondisi	ident_where_tabel[1] : category ident_where_tabel[2] : film	
[Keluaran] Identifikasi Atribut Kondisi	ident_where_atribut[1] : name ident_where_atribut[2] : description	
[Keluaran] Identifikasi Operator Banding	ident_where_banding[1] : = ident_where_banding[2] : LIKE	
[Keluaran] Identifikasi Isi Kondisi Operator Logika	ident_where_isi[1] : 'comedy' ident_where_isi[2] : '%dentist%'	

Pada Tabel 11 ditemukan 2 keyword fitur WHERE, dan dilakukan pengecekan kebutuhan fitur, sehingga syarat untuk menggunakan fitur ini terpenuhi.

e. Penyusunan Query

Tahap ini adalah tahap terakhir dari proses terjemahan, dan akan dilakukan pengecekan hasil dari setiap proses identifikasi sebelumnya.



Gambar 13. Alur Proses Penyusunan Query

Pada proses ini pengecekan dilakukan pada setiap hasil identifikasi, jika ada hasil identifikasi yang tidak sesuai maka akan membuat hasil query tidak valid dan untuk beberapa pengecekan akan muncul pesan eror.

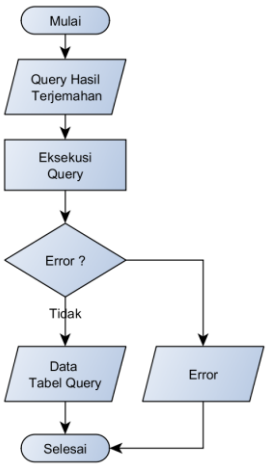
Tabel 12. Hasil Proses Identifikasi Fitur

Nama Objek	Objek
[Masukan] Identifikasi Perintah	Ident_perintah : SELECT
[Masukan] Identifikasi Tabel	ident_table[1] : film ident_table[2] : category ident_relasi[1] : FROM `film` ident_relasi[2] : INNER JOIN `film_category` ON `film`.`film_id` = `film_category`.`film_id` ident_relasi[3] : INNER JOIN `category` ON `film_category`.`category_id` = `category`.`category_id`
[Masukan] Identifikasi Atribut	ident_attribute[1] : `title` AS 'judul' ident_attribute[2] : `description` AS 'deskripsi'
[Masukan] Identifikasi Kondisi	ident_where[1] : yang ident_where[2] : dimana ident_where_logika[1] : AND ident_where_tabel[1] : category ident_where_tabel[2] : film ident_where_atribut[1] : name ident_where_atribut[2] : description ident_where_banding[1] : = ident_where_banding[2] : LIKE ident_where_isi[1] : 'comedy' ident_where_isi[2] : '%dentist% [11]
[Keluaran] Penyusunan Query	query_part[1] : SELECT query_part[2] : `film`.`title` AS 'judul' query_part[3] : FROM `film` query_part[4] : INNER JOIN `film_category` ON `film`.`film_id` = `film_category`.`film_id` query_part[5] : INNER JOIN `category` ON `film_category`.`category_id` = `category`.`category_id` query_part[6] : WHERE query_part[7] : `category`.`name` = 'comedy' query_part[8] : AND query_part[9] : `film`.`description` LIKE '%dentist%' query_part[10] : ;
[Keluaran] Hasil Query	SELECT `film`.`title` AS 'judul' FROM `film` INNER JOIN `film_category` ON `film`.`film_id` = `film_category`.`film_id` INNER JOIN `category` ON `film_category`.`category_id` = `category`.`category_id` WHERE `category`.`name` = 'comedy' AND `film`.`description` LIKE '%dentist%';

Pada Tabel 12 tidak ditemukan kesahajaan identifikasi.

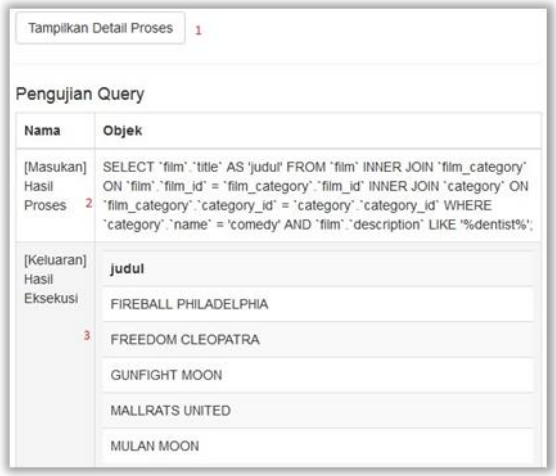
3.4 Pengujian Query Hasil

Pada Proses terakhir ini adalah menguji hasil yang sudah di terjemahkan ke dalam syntax SQL untuk dieksekusi syntaxnya dan dilihat data yang menjadi ouputnya.



Gambar 14. Alur Proses Pengujian Query Hasil

Jika query hasil terjemahan tidak valid maka akan menampilkan pesan eror.



Gambar 15. Alur Proses Pengujian Query Hasil

Pada Gambar 15 menujukan hasil eksekusi dari query hasil terjemahan.

4. PENGUJIAN

Pengujian ini dilakukan pengkategorian tipe-tipe query yang akan diuji, berikut adalah kategori query :

Tabel 13. Tipe Query

No	Tipe Query
1	Query tanpa kondisi
2	Query dengan kondisi
3	Query Banyak kondisi
4	Query banyak tabel tanpa kondisi
5	Query banyak tabel banyak kondisi
6	Query dengan order

Adapun kalimat yang akan diuji yang telah dikelompokkan berdasarkan kategorinya adalah sebagai berikut :

Tabel 14. Kalimat Pengujian

No	Kalimat	Tipe	Query yang diharapkan
1	Tampilkan mahasiswa	1	select * from mahasiswa
2	Tampilkan nama mahasiswa	1	select nama from mahasiswa
3	Lihat nilai uas dan uts	1	select uas , uts from nilai
4	sebutkan nama, alamat dan kota mahasiswa	1	select nama, alamat, kota from mahasiswa
5	Lihat mahasiswa yang nimnya 10113005	2	select * from mahasiswa where nim = '10113005'
6	tampilkan nim mahasiswa yang namanya jamal	2	select nim from mahasiswa where nama = 'jamal'
7	tampilkan nama mahasiswa yang jenis kelaminnya perempuan	2	select nama from mahasiswa where jk = 'perempuan'
8	tampilkan mahasiswa berdasarkan jenis kelamin	6	select nama from mahasiswa order by jk
9	tampilkan nilai uas diurutkan berdasarkan paling kecil	6	select * from mahasiswa where uas sort by ascending
10	lihat mahasiswa yang tinggal di bandung dan namanya ahmad	3	select * from mahasiswa where kota = 'bandung' and nama = ahmad
11	tampilkan mahasiswa yang berjenis kelamin perempuan dan tinggal di bandung	3	select * from mahasiswa where jk = 'perempuan' and kota = 'bandung'
12	tampilkan nilai dan kelas	4	select * from nilai join kelas on nilai.kode_kelas = kelas.kode_kelas
13	lihat kelas mahasiswa	4	select * from mahasiswa join kelas on mahasiswa.kode_kelas = kelas.kode_kelas
14	sebutkan nama mahasiswa dan nilai uas	4	select nama , uas from mahasiswa join nilai on mahasiswa.nim = nilai.nim
15	sebutkan nama kelas dan nilai uas	4	select nama_kelas, uas from nilai join kelas on nilai.kode_kelas = kelas.kode_kelas
16	sebutkan mahasiswa yang nilai uasnya 80 sampai 90	5	select * from mahasiswa join nilai on mahasiswa.nim = nilai.nim where uas between 80 and 90

No	Kalimat	Tipe	Query yang diharapkan
17	siapa mahasiswa yang uas nya bernilai 90	5	select * nama from mahasiswa join nilai on mahasiswa.nim = nilai.nim where uas = '90'
18	Tampilkan judul film yang termasuk kategori comedy, dimana dalam deskripsi film tersebut mengandung kata "dentist"	5	SELECT title AS "Judul Film" FROM film JOIN film_category USING (film_id) JOIN category USING (category_id) WHERE category.name='comedy' AND description LIKE '%dentist%'

Hasil dari pengujian adalah sebagai berikut :

Tabel 15. Pengujian Kalimat

No	Hasil Pengujian	Kesimpulan
1	SELECT * FROM `mahasiswa` ;	Sesuai
2	SELECT `mahasiswa`.`nama` FROM `mahasiswa` ;	Sesuai
3	SELECT `nilai`.`uas` , `nilai`.`uts` FROM `nilai` ;	Sesuai
4	SELECT `mahasiswa`.`nama` , `mahasiswa`.`alamat` , `mahasiswa`.`kota` FROM `mahasiswa` ;	Sesuai
5	SELECT * FROM `mahasiswa` WHERE `mahasiswa`.`nim` = '10113005';	Sesuai
6	SELECT `mahasiswa`.`nim` FROM `mahasiswa` WHERE `mahasiswa`.`nama` = 'jamal';	Sesuai
7	SELECT `mahasiswa`.`nama` FROM `mahasiswa` WHERE `mahasiswa`.`jk` = 'perempuan';	Sesuai
8	tampilkan mahasiswa berdasarkan jenis kelamin	Sesuai
9	SELECT `nilai`.`uas` FROM `nilai` ORDER BY ``.`` DESC;	Kurang sesuai
10	SELECT * FROM `mahasiswa` WHERE `mahasiswa`.`kota` = 'di';	Kurang sesuai
11	SELECT * FROM `mahasiswa` WHERE `mahasiswa`.`kota` = 'di';	Kurang sesuai
12	SELECT * FROM `nilai` INNER JOIN `kelas` ON `nilai`.`kode_kelas` = `kelas`.`kode_kelas` ;	Sesuai
13	SELECT * FROM `kelas` INNER JOIN `nilai` ON `kelas`.`kode_kelas` = `nilai`.`kode_kelas` INNER JOIN `mahasiswa` ON `nilai`.`nim` = `mahasiswa`.`nim` ;	Sesuai
14	SELECT `mahasiswa`.`nama` , `nilai`.`uas` FROM `mahasiswa` INNER JOIN `nilai` ON `mahasiswa`.`nim` = `nilai`.`nim` ;	Sesuai
15	SELECT `kelas`.`nama_kelas` AS 'nama' , `nilai`.`uas` FROM `kelas` INNER JOIN `nilai` ON `kelas`.`kode_kelas` = `nilai`.`kode_kelas` ;	
16	SELECT * FROM `mahasiswa` INNER JOIN `nilai` ON `mahasiswa`.`nim` = `nilai`.`nim` WHERE `nilai`.`uas` = '80';	Kurang sesuai
17	SELECT * FROM `mahasiswa` WHERE ;	Tidak sesuai
18	SELECT `film`.`title` AS 'judul' FROM `film` INNER JOIN `film_category` ON `film`.`film_id` = `film_category`.`film_id` INNER JOIN `category` ON `film_category`.`category_id` = `category`.`category_id` WHERE `category`.`name` = 'comedy' AND `film`.`description` LIKE '%dentist%';	Sesuai

Pengujian akurasi yang digunakan dalam penelitian ini menerapkan metode F – Measure. F-Measure adalah representasi ukuran keakuratan dari sebuah tes dengan menggunakan nilai precision dan recall dari tes tersebut [7].

Pengujian ini melihat apakah sistem dapat memproses kalimat yang diinputkan oleh pengguna agar dapat diterjemahkan sehingga dapat diukur akurasinya dengan metode F – Measure. kalimat yang diinputkan telah diterjemahkan secara manual terlebih dahulu sehingga dapat dibandingkan dengan hasil keluaran dari sistem. Nilai – nilai yang dihitung pada pengujian ini adalah nilai correct, falsePositive, falseNegative, precision, recall, dan F – Measure dari iterasi dataset sistem. Perhitungan F – Measure microaverage data dengan menggunakan total nilai correct, falsePositive, falseNegative keseluruhan dataset sedangkan perhitungan sedangkan F – Measure macroaverage adalah dengan menghitung rata – rata F – Measure setiap iterasi dataset. Berikut merupakan hasil dari iterasi seluruh dataset pengujian akurasi sistem.

$$\begin{aligned} &Dataset\ Kalimat \\ &Total\ Dataset = TD = 18 \\ &Total\ nilai\ correct = Tnc = 13 \\ &total\ nilai\ falsePositive = tnfP = 4 \\ &total\ nilai\ falseNegative = tnfN = 1 \end{aligned}$$

$$Precision_{micro} = \frac{Tnc}{(Tnc + tnfP)} = \frac{13}{(13+4)} = \frac{13}{17} = 0.7647 \quad (1)$$

$$Recall_{micro} = \frac{Tnc}{(Tnc + tnfN)} \quad (2)$$

$$= \frac{13}{(13+1)} = \frac{13}{14} = 0.9286$$

$$F-Measure_{microaverage} = \frac{2\ x\ \left(Precision_{micro}\right)\ x\ \left(Recall_{micro}\right)}{\left(Precision_{micro}\right) + \left(Recall_{micro}\right)} \quad (3)$$

$$= \frac{2\ x\ (0.7647)\ x\ (0.9286)}{(0.7647) + (0.9286)} = \frac{1.4202}{1.6933}$$

$$= 0.8387\ X\ 100\% = 83.87\%$$

$$F-Measure_{microaverage} = \sum_{i=0}^{n=40} F-Measure_{microaverage}\ x100\% \quad (4)$$

$$= \frac{1+1+0+0+\cdots+0+1+1}{40} x\ 100\%$$

$$= \frac{13}{18} x\ 100\% = 0.7222\ x\ 100\% = 72.22\%$$

Jadi akurasi sistem berdasarkan dataset pengujian diatas adalah: Nilai keakuratan diketahui berkisar antara 72.22% - 83.87%.

5. PENUTUP

3.1 Kesimpulan

Setelah melakukan analisis, perancangan, dan pengujian dengan metode blackbox dan pengujian akurasi, diperoleh informasi bahwa sistem dapat menerjemahkan kalimat perintah dalam bahasa alami ke dalam format syntax query SQL. Hasil uji dari beberapa kalimat didapatkan akurasi berkisar 72.22% - 83.87%. 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

Untuk pengembangan penelitian ke arah yang lebih baik, maka terdapat beberapa saran untuk meningkatkan kemampuan sistem dan menambahkan fitur-fitur lain sebagai berikut :

- Jadikan sistem sebagai library bukan aplikasi dengan Tools analisis dan form alias dijadikan sebuah konfigurasi untuk digunakan di aplikasi lain.
- Penambahan fitur untuk mengatasi kata yang yang berkaitan seperti nama tempat, nama lengkap, suatu kalimat, tanggal dan sebagainya.
- Penambahan fitur fungsi Agregasi, DATE TIME, DISTINCT, GROUP BY, HAVING, LIMIT, SUB QUERY, ekspresi pada SELECT.
- Penambahan rule untuk relasi selain INNER JOIN, rule untuk fitur LIKE yang lebih dinamis, seperti berawalan sekian huruf dan lain-lain, rule untuk validasi-validasi kesalahan yang mungkin terjadi, rule penerjemahan kalimat ke query SQL agar dapat meningkatkan akurasi kunci jawaban sistem.
- Penambahan soal Uji yang lebih variatif untuk menguji segala kemungkinan.
- Pengembangan dalam cakupan lebih luas seperti ranah DML, DDL, DCL, dan sebagainya

DAFTAR PUSTAKA

[1] S. Wibisono, Aplikasi Pengolahan Bahasa Alami untuk Query Basisdata Akademik dengan Format Data XML, Skripsi S1. Teknik Informatika. Universitas Stikubank, 2013.

[2] N. Isharmawan, Ekstraksi Informasi dan SQL Parser untuk Query SQL dari Teks Berbahasa Indonesia, Skripsi S1. Teknik Informatika. Universitas Komputer Indonesia, 2017.

[3] M. H. Taqwim, Ekstraksi Informasi dengan Metode Rule – Based untuk Evaluasi Pemahaman Fisika Kinematika, Skripsi S1. Teknik Informatika. Universitas Komputer Indonesia, 2016.

[4] S. Entin, "Kecerdasan Buatan," 28 September 2011. [Online]. Available: http://entin.lecturer.pens.ac.id/Kecerdasan%20Buatan/Buku/. [Accessed 19 April 2017].

[5] A. Ismaya, Algoritma Ekstraksi Informasi Berbasis Aturan, Yogyakarta: Skripsi S1. Teknik Informatika : Universitas Gajah Mada, 2014.

[6] M. L. Khodra, Ekstraksi Informasi Transaksi Online pada Twitter, Bandung: Skripsi S1. Teknik Informatika : Institut Teknologi Bandung, 2013.

[7] MySQL Documentation "Sakila Sample Database,". [Online]. Available: https://dev.mysql.com/doc/sakila/en/ [Accessed 24 Februari 2018].

[8] H. Ruwdanar, R. Arum SP. Dan H. Fitriawan, Rancang Bangun Sistem Informasi Akademik Menggunakan Php Dan Mysql, Bandar Lampung: Universitas Negeri Lampung, 2012.