



High Education Of Organization Archive Quality

**PERANCANGAN PENJADWALAN KULIAH STIKOM UYELINDO DENGAN ALGORITMA GENETIK**

M.I.J. Lamabelawa, Bruno Sukarto

**KLASIFIKASI KINERJA BEBAN DOSEN STIKOM UYELINDO KUPANG MENGGUNAKAN ALGORITMA DECISION TREE**

Petrus Katemba

**REPLIKASI BASIS DATA PADA SISTEM INFORMASI ONLINE**

Marlina

**PEMBANGUNAN SISTEM KEHADIRAN SISWA BERBASIS SHORT MESSAGE SERVICE (STUDI KASUS SMK UYELINDO KUPANG)**

Max ABR Soleman Lenggu

**PERAMALAN PERTUMBUHAN JUMLAH CALON MAHASISWA BARU PADA STIKOM UYELINDO KUPANG MENGGUNAKAN EXPONENTIAL SMOOTHING**

Tri Ana Setyarini

**SISTEM Pencarian PASAL-PASAL PERATURAN DAERAH PADA SEKRETARIAT DAERAH KOTA KUPANG BERBASIS WEB**

Dewi Anggraini

**APLIKASI PENGENALAN ANGKA DAN WARNA DALAM BAHASA INGGRIS PADA MURID SEKOLAH DASAR**

Heni, Skolastika Siba Igon

**APLIKASI PENGOLAHAN DATA OBAT PADA APOTIK SARI KURNIA KUPANG BERBASIS CLIENT SERVER.**

Mardhalia Saitakela

**PENERAPAN QUICK RESPONSE CODE UNTUK KARTU NAMA PADA SMARTPHONE BERBASIS ANDROID**

Mario, Sri Andayani

**CHITCATCINEMA: APLIKASI QUESTION ANSWERING SYSTEM UNTUK DOMAIN FILM BIOSKOP**

Ryan Agus Setiawan, R. Kristoforus J. Bendi

**PENENTUAN LOKASI HOTEL DI KOTA KUPANG BERBASIS GEOGRAPHIC INFORMATION SYSTEMS**

Remerta Naatonis

**EVALUASI KINERJA DOSEN PEMBIMBING SKRIPSI (STUDI KASUS: STIKOM UYELINDO KUPANG)**

Donna Setiawati

**IMPLEMENTASI SISTEM ADMINISTRASI KELURAHAN KAYU PUTIH BERBASIS MULTI USER**

Yohanes Payong

**IDENTIFIKASI JENIS KAYU BERDASARKAN CIRI UMUM DENGAN PENDEKATAN GOAL DRIVEN**

Yohanes Suban Belutowe

## DEWAN REDAKSI

**Pelindung** : Ketua STIKOM UYELINDO KUPANG

**Penanggung Jawab** : Wakil Ketua Bidang Tridarma

Ketua Lembaga Penelitian, Publikasi dan Pengembangan pada Masyarakat

### **Penyunting Ahli/Mitra Bestari:**

1. Prof. Ir. Suyoto, Ph.D (Univ. Atma Jaya Yogyakarta)
2. Prof. Dr. Ir. Eko Sedyono, M.Kom (UKSW Salatiga)
3. Prof. Ir. Daniel Manongga, M.Sc. Ph.D (UKSW Salatiga)
4. Prof. Dr. Ir. Kuswara Setiawan, MT (UPH)
5. Drs. Siprianus Garak, M.Sc (UNDANA Kupang)

### **Penyunting Pelaksana:**

1. Max ABR. Soleman Lenggu, S.Kom, MT
2. Marinus I.J. Lamabelawa, S.Kom, M.Cs
3. Semlinda J Bulan, M.Kom.

### **Alamat Sekretariat/Redaksi:**

Lembaga Penelitian, Publikasi dan Pengembangan pada Masyarakat  
STIKOM Uyelindo Kupang  
Jl. Perintis Kemerdekaan 1, Kayu Putih, Kupang, NTT, Indonesia.  
Telp. (0380)8554501, Fax (0380)  
Email : lp3muyelindo@gmail.com  
<http://www.uyelindo.ac.id>

**Jurnal Hoaq** merupakan Jurnal Ilmiah untuk menampung hasil penelitian yang berhubungan dengan bidang sains dan teknologi. Bidang penelitian yang dimaksud adalah Soft Computing, Mobile Computing, dan Rekayasa Perangkat Lunak.

**Jurnal Hoaq** diterbitkan oleh Lembaga Penelitian, Publikasi dan Pengembangan pada Masyarakat, Bekerja sama dengan Program Studi Teknik Informatika dan Program Studi Sistem Informasi STIKOM Uyelindo Kupang. **Redaksi** mengundang para profesional dari dunia usaha, pendidikan dan peneliti untuk menulis mengenai perkembangan ilmu di bidang **Teknologi Informasi**.

**Jurnal Hoaq** diterbitkan 2(dua) kali dalam 1(satu) tahun pada bulan Mei dan Desember

## DAFTAR ISI

	Halaman
PERANCANGAN PENJADWALAN KULIAH STIKOM UYELINDO DENGAN ALGORITMA GENETIK M.I.J. Lamabelawa, Bruno Sukarto	152-159
KLASIFIKASI KINERJA BEBAN DOSEN STIKOM UYELINDO KUPANG MENGGUNAKAN ALGORITMA DECISION TREE Petrus Katemba	160-165
REPLIKASI BASIS DATA PADA SISTEM INFORMASI ONLINE Marlina	166-169
PEMBANGUNAN SISTEM KEHADIRAN SISWA BERBASIS SHORT MESSAGE SERVICE (STUDI KASUS SMK UYELINDO KUPANG) Max ABR Soleman Lenggu	170-176
PERAMALAN PERTUMBUHAN JUMLAH CALON MAHASISWA BARU PADA STIKOM UYELINDO KUPANG MENGGUNAKAN EXPONENTIAL SMOOTHING Tri Ana Setyarini	177-184
SISTEM Pencarian PASAL-PASAL PERATURAN DAERAH PADA SEKRETARIAT DAERAH KOTA KUPANG BERBASIS WEB Dewi Anggraini	185-191
APLIKASI PENGENALAN ANGKA DAN WARNA DALAM BAHASA INGGRIS PADA MURID SEKOLAH DASAR Heni, Skolastika Siba Igon	192-199
APLIKASI PENGOLAHAN DATA OBAT PADA APOTIK SARI KURNIA KUPANG BERBASIS CLIENT SERVER. Mardhalia Saitakela	200-204
PENERAPAN <i>QUICK RESPONSE CODE</i> UNTUK KARTU NAMA PADA <i>SMARTPHONE</i> BERBASIS <i>ANDROID</i> Mario, Sri Andayani	205-210
<i>CHITCATCINEMA: APLIKASI QUESTION ANSWERING SYSTEM</i> UNTUK DOMAIN FILM BIOSKOP Ryan Agus Setiawan, R. Kristoforus J. Bendi	211-218
PENENTUAN LOKASI HOTEL DI KOTA KUPANG BERBASIS GEOGRAPHIC INFORMATION SYSTEMS Remerta Naatonis	219-225
EVALUASI KINERJA DOSEN PEMBIMBING SKRIPSI (STUDI KASUS: STIKOM UYELINDO KUPANG) Donna Setiawati	226-231
IMPLEMENTASI SISTEM ADMINISTRASI KELURAHAN KAYU PUTIH BERBASIS MULTI USER Yohanes Payong	232-240
IDENTIFIKASI JENIS KAYU BERDASARKAN CIRI UMUM DENGAN PENDEKATAN <i>GOAL DRIVEN</i> Yohanes Suban Belutowe	241-249

## ***CHITCATCINEMA: APLIKASI QUESTION ANSWERING SYSTEM UNTUK DOMAIN FILM BIOSKOP***

**Ryan Agus Setiawan<sup>1</sup>, R. Kristoforus J. Bendi<sup>2</sup>**

*Teknik Informatika, Sekolah Tinggi Teknik Musi*

*E-mail: 25minutes.again@gmail.com<sup>1</sup>, kristojb@sttmusi.ac.id<sup>2</sup>, andayani\_s@sttmusi.ac.id*

### **ABSTRACT**

*Along with the development of times, informations stored on Internet will spread more widely. This is certainly beneficial for cinema fans who visit sites like 21cineplex and IMDb . However, it is not matched by the ease in finding desired information. Users need to browse web pages for further information desired. The site also displays a lot of images that resulted in long waiting times, big internet traffic consumption, heavy and impressed when accessed via mobile phone.*

*Based on these problems, then application named Chit Chat Cinema was built. This application is an implementation of the Question Answering System where users simply enter question by using Indonesian language and it will give an appropriate answer. Questions that can be recognized is the title of movie based on a particular actor, now playing movie, coming soon movie, duration, actor's name whose played by particular movie title, genre, and synopsis. This application is targeted for Android platform. Applications connect to server that can respond and provide answers to user. Web service that located on server created using NodeJS and MySQL database. Information about movie was collected by using screen scraping to 21cineplex site. Question processing methods are tokenization, part-of-speech tagging, and rule-based named entity recognition. Process model is sequential linear.*

*From 260 questions were tested, there are 208 questions or 80% of total questions can answered correctly. The system gives wrong answer if question use unknown pattern, contain words that aren't recognized, and the name of entity is an Indonesian loan word.*

**Keywords:** *Question Answering, Scraping, Web Service, Android, Film, Cinema, 21cineplex, Waterfall*

## 1. PENDAHULUAN

Seiring dengan berkembangnya zaman, informasi yang tersimpan pada internet juga akan berkembang semakin luas. Informasi ini tentunya sangat bermanfaat bagi sebagian orang, tak terkecuali bagi para penggemar film bioskop. Mereka menggunakan internet untuk mengetahui informasi di beberapa situs seperti 21cineplex dan IMDb.

Informasi di internet dapat diakses melalui berbagai jenis perangkat, mulai dari komputer, laptop, *smartphone*, hingga tablet. Lembaga riset Walker Sands, melaporkan pada kuartal ketiga di tahun 2013, akses internet yang dilakukan langsung melalui perangkat *mobile* tercatat mengalami peningkatan sebesar 67% dibandingkan pada kuartal yang sama di tahun 2012. Secara total, keseluruhan trafik internet pada kuartal ketiga tahun 2013 menyumbang pertumbuhan sebesar 28% [3].

Namun semakin berkembangnya informasi di dunia maya tidak diimbangi dengan kemudahan dalam mencari informasi yang diinginkan. Berdasarkan survei awal yang dilakukan pada tanggal 24 Maret 2013 dengan melibatkan 16 responden, beberapa pengguna mengalami kesulitan saat mengunjungi situs film bioskop seperti 21cineplex, terutama menggunakan ponsel. Pengguna perlu menelusuri lebih lanjut pada halaman web untuk mendapatkan informasi yang diinginkan. Selain itu kenyamanan pengguna juga ikut terganggu karena menampilkan banyak gambar yang mengakibatkan waktu tunggu yang cukup lama, jumlah akses data Internet yang relatif besar, dan terkesan berat untuk diakses.

Berdasarkan permasalahan di atas, salah satu solusi yang bisa ditawarkan adalah *Question Answering System* (QAS). Melalui QAS, pengguna dapat memasukkan pertanyaan dengan menggunakan bahasa alami dan komputer dapat memahami pertanyaan yang dimaksud, kemudian aplikasi akan memberikan jawaban yang sesuai [8]. Hal ini membuat pengguna mendapatkan informasi secara langsung dibandingkan dengan menggunakan mesin pencarian atau membuka website seperti 21cineplex, yang mengharuskan pengguna untuk melakukan navigasi ke konten yang diinginkan.

Sistem ini diharapkan dapat membantu pengguna perangkat *mobile* dalam mencari informasi yang dibutuhkan secara cepat serta dengan interaksi yang senatural mungkin, khususnya masyarakat Indonesia yang kesulitan dalam menggunakan aplikasi dan memiliki keterbatasan dalam mengakses internet, baik dari sisi kecepatan maupun jumlah kuota yang tersedia. Aplikasi yang dibuat dapat mengenali pertanyaan yang menggunakan bahasa Indonesia, karena bahasa Indonesia merupakan bahasa resmi

negara Indonesia dan aplikasi ditujukan untuk masyarakat Indonesia.

## 2. TINJAUAN PUSTAKA

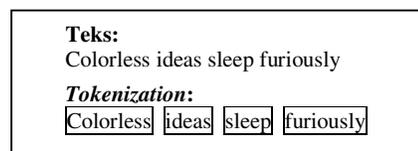
### 2.1 Question Answering System

*Question Answering System* (QAS) merupakan sebuah sistem yang memungkinkan komputer dapat memahami maksud pertanyaan yang diajukan oleh pengguna dengan menggunakan bahasa alami dan merespon pertanyaan tersebut berdasarkan informasi yang ada [8]. QAS dikelompokkan menjadi dua bagian, yaitu QAS dengan domain terbuka dan QAS dengan domain tertutup. QAS dengan domain terbuka memungkinkan sistem dapat menjawab pertanyaan yang sifatnya umum. Sedangkan QAS dengan domain tertutup hanya dapat menjawab pertanyaan dengan topik khusus, misalnya tentang musik, cuaca, dan buku.

Supaya dapat memahami maksud dari pertanyaan pada QAS, tahap *preprocessing* diperlukan untuk mengekstrak informasi suatu teks. Ada beberapa metode yang digunakan dan dapat dikombinasikan sesuai kebutuhan. Dalam penelitian ini akan menggunakan *preprocessing* pada teks pertanyaan dengan metode *tokenization*, *POS tagging*, dan *Named Entity Recognition*.

### 2.2 Tokenization (tokenisasi)

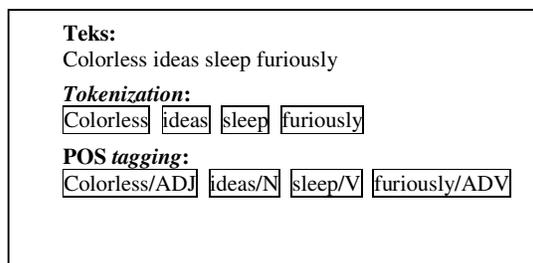
*Tokenization* adalah proses pembagian sebuah teks menjadi unit dasar (*token*). Pada umumnya *token* dipisahkan berdasarkan spasi dan tanda baca, sehingga setiap *token* diwakili dalam bentuk kata atau tanda baca [4]. Hasil dari *tokenization* dapat dilihat pada Gambar 1.



Gambar 1. *Tokenization* Pada Teks

### 2.3 Part-of-Speech Tagging

*Part-of-Speech* (POS) *tagging* adalah proses mengklasifikasikan dan melabelkan suatu kata ke dalam kelompok *part-of-speech* [4]. *Part-of-speech* adalah kumpulan tata bahasa yang terdiri dari kata kerja, kata benda, kata sifat, dan lain-lain. Proses ini dilakukan setelah *tokenization*. Proses POS *tagging* dapat dilihat pada Gambar 2.



Gambar 2. Pemberian Label POS

## 2.4 Named Entity Recognition

Menurut Nugroho dalam [2], *Named Entity Recognition* (NER) memiliki peran yang cukup vital karena tugasnya adalah mengenali entitas pada teks. Contohnya adalah entitas nama (nama orang, lokasi, organisasi), entitas waktu (tanggal, waktu, durasi), serta entitas bilangan (uang, persen, numerik, kardinal). Sedangkan entitas itu sendiri adalah objek di dunia nyata yang dapat dibedakan dari yang lainnya.

Salah satu cara untuk mengenali entitas dapat dilakukan dengan menggunakan pendekatan *rules-based*. Pendekatan ini menggunakan aturan-aturan yang dibuat secara manual untuk mengenali entitasnya. Kinerjanya bergantung pada pola-pola yang ditemukan pada dokumen observasi. Pendekatan ini cocok digunakan untuk teks yang singkat dan memiliki pola yang mudah dikenali.

Pengenalan pola berbasis *rules-based* dapat dilakukan berdasarkan ciri tertentu seperti penggunaan huruf kapital, tipe POS yang digunakan untuk *token* sekitarnya, serta *token* yang diawali atau diakhiri dengan kata tertentu. NER pada penelitian ini juga dapat mendeteksi kumpulan kata yang dianggap sebagai nama objek seperti pada pertanyaan “Siapakah aktor dari film The Lord of the Ring?”. Hasil yang diharapkan dari pertanyaan itu adalah *string* “The Lord of the Ring” dianggap sebagai entitas film.

## 2.5 JavaScript Object Notation (JSON)

JSON merupakan format pertukaran data yang menggunakan notasi literal objek dari bahasa JavaScript. JSON sejatinya adalah format teks sehingga dapat dibaca oleh manusia maupun mesin. JSON memiliki 6 jenis nilai, yaitu: objek, *array*, *string*, angka, *boolean* (*true* dan *false*), dan *null* [5].

Objek dari JSON adalah sepasang *key* dan *value* yang tidak berurutan. Sebuah *key* berbentuk *string*, sedangkan *value* bisa berupa nilai JSON, termasuk *array* dan objek. Tujuan dari desain JSON adalah minimal, *portable*, tekstual, dan merupakan *subset* dari JavaScript.

## 2.6 Web Service

*Web service* adalah salah satu bentuk sistem perangkat lunak yang didesain untuk mendukung interaksi mesin-ke-mesin melalui jaringan. *Web service* memiliki *interface* yang dideskripsikan dalam format yang dapat dibaca oleh mesin [9].

*Web service* dibagi menjadi dua jenis, yaitu REST dan SOAP. REST (*REpresentational State Transfer*) lebih banyak digunakan untuk *web service* yang berorientasi pada *resource* [9]. *Resource* yang dimaksud adalah data-data yang disediakan saat mengakses *web service*. *Resource* dapat dilihat dan dimanipulasi melalui protokol HTTP. Contohnya adalah *web service* dari situs Twitter, di mana *developer* dapat membuat aplikasi yang dapat menampilkan *tweet* pengguna, mengirimkan *tweet*, atau menghapus *tweet* yang berasal dari *account* sendiri.

## 2.7 Web Scraping

*Website* yang selama ini kita kenal dirancang untuk berinteraksi dengan manusia. Sedangkan *web service* dibangun agar *website* dapat berinteraksi dengan mesin. Namun tidak semua pengelola *website* membangun *web service*. Oleh karena itu, dibutuhkan cara untuk menggali informasi dari *website* yang ditujukan untuk manusia agar informasi tersebut juga dapat dimengerti oleh mesin. Proses ini disebut *web scraping* [12].

Tugas dari *web scraping* adalah mentransformasikan isi suatu web yang tidak terstruktur, dalam hal ini adalah format HTML, menjadi format data yang lebih terstruktur yang nantinya dapat disimpan ke basis data. Pada penelitian ini, *web scraping* digunakan untuk mengekstrak isi dari situs portal film.

Untuk melakukan *web scraping*, hal pertama yang dilakukan adalah men-*download* file HTML yang berisi informasi yang dibutuhkan. Kemudian, struktur HTML tersebut perlu dianalisa untuk mengetahui lokasi informasi yang dibutuhkan. Selanjutnya, lakukan *parsing* untuk memisahkan suatu teks dari tag HTML pada halaman tersebut. Proses *parsing* dapat dilakukan melalui *XPath* ataupun *CSS Selector*. Penelitian ini akan menggunakan *CSS Selector* sebagai *parser* dari halaman web.

## 2.8 Penelitian Terdahulu

Penelitian mengenai QAS yang memanfaatkan informasi di Internet baik domain terbuka maupun domain tertutup telah dilakukan oleh beberapa peneliti. Chung *et al* [6] membangun QAS dengan domain peramalan cuaca. Informasi mengenai peramalan cuaca didapatkan dengan melakukan *crawling* terhadap situs peramalan cuaca dari Korea Selatan dan informasi tersebut disimpan ke dalam

database. Teks pertanyaan yang dikenali adalah berbahasa Inggris. Tahapan *preprocessing* teks yang digunakan adalah *Named Entity Recognition*, *Temporal Data Normalizer*, dan *Inference Rules*.

Ferrández *et al* [7] membangun QACID yang hanya dapat menjawab pertanyaan berhubungan dengan film bioskop. Tahapan *preprocessing* dari QACID adalah *morphological analyzer* dan *NER*. Bahasa yang dikenali pada aplikasi tersebut adalah bahasa Spanyol.

YAGO-QA yang dibangun oleh Adolphs *et al* [1] dapat merespon hampir semua pertanyaan dengan *knowledge base* berasal dari YAGO (RDF dengan *knowledge base*-nya diekstrak dari Wikipedia). Bahasa yang dikenalnya adalah bahasa Inggris dan tahapan *preprocessing* dimulai dari *tokenization*, *morphology*, dan *NER*.

Srihari dan Li [10] mengembangkan QAS yang dapat mengekstrak informasi mengenai kejadian umum. Sumber informasi diperoleh dari kumpulan teks narasi berbahasa Inggris. Metode *preprocessing* yang digunakan adalah *Tokenization*, *POS tagging*, *Shallow Parsing*, dan *NER*.

Unger *et al* [11] dalam penelitian *Template-based Question Answering over RDF Data* membangun sebuah sistem QA berbahasa Inggris. Domain yang digunakan bersifat terbuka di mana *knowledge base* yang diambil berasal dari RDF DBpedia. Sedangkan tahapan *preprocessing* yang digunakan adalah *POS tagging*, *Semantic Representation*, dan *NER*.

Penelitian ini berfokus pada membangun sebuah sistem QA dengan domain film bioskop. *Knowledge base* yang digunakan berasal dari hasil *web scraping* situs *21cinplex*. Tahapan *preprocessing* dimulai dari *tokenization*, *POS tagging*, dan *NER*. Sistem QA yang dibangun nantinya dapat mengenali pertanyaan berbahasa Indonesia.

### 3. METODE PENELITIAN

#### 3.1 Analisis Situs 21cinplex

Halaman yang memuat daftar film akan dilakukan *scraping* untuk mendapatkan detail film. Setelah struktur HTML tersebut diteliti lebih lanjut, maka *CSS Selector* yang digunakan dapat dilihat pada Tabel 1. Namun ada beberapa film tidak menampilkan nama aktor, kategori film, dan durasi film dari situs *21cinplex*.

Tabel 1. Informasi Daftar Film Beserta CSS Selectornya

No	Informasi yang dicari	CSS Selector
1	Sinopsis film	<i>Head meta [name="description"] [content]</i>
2	Nama aktor	<i>.cast ul li p</i>
3	URL foto aktor	<i>.cast ul li img[src]</i>
4	Kategori atau rating film	<i>.movinfo [title]</i>

5	Durasi film	<i>.duration h3</i>
---	-------------	---------------------

Selanjutnya adalah mengumpulkan daftar film yang akan tayang dari URL <http://21cinplex.com/comingsoon>. Struktur halaman ini menyerupai struktur halaman daftar film. *CSS Selector* yang digunakan seperti pada Tabel 2.

Tabel 2. Film Coming Soon dan CSS Selectornya

No.	Informasi yang dicari	CSS Selector
1	Judul film	<i>.col-m_462 .col-content ul li a img[title]</i>
2	URL gambar poster film	<i>.col-m_462 .col-content ul li a img[src]</i>
3	URL link More	<i>.col-m_462 .col-content ul li p a[href]</i>

Daftar film yang sedang tayang dapat dilihat melalui URL <http://21cinplex.com/theater>. Film yang akan tayang akan tampil saat nama bioskop dipilih. Hal ini dikarenakan setiap bioskop menampilkan informasi yang berbeda untuk daftar film yang akan tayang.

Penelitian ini lebih berfokus pada informasi di bioskop PIM XXI. Oleh karena itu, halaman yang dicari untuk menampilkan daftar film *coming soon* berada di bagian halaman khusus bioskop PIM XXI dan didapatkanlah URL <http://www.21cinplex.com/theater/palembang,242,PLGPIM.htm>. Di halaman ini terdapat tabel yang menampilkan judul film yang sedang tayang beserta jam tayang dan kategori film. Berdasarkan hasil analisa struktur HTML didapatkan lokasi DOM yang dibutuhkan seperti pada Tabel 3.

Tabel 3. Film Coming Soon Beserta CSS Selectornya

No.	Informasi yang dicari	CSS Selector
1	Judul film	<i>.table-theater-det tr td a:odd</i>
2	URL gambar poster film	<i>.table-theater-det tr td a:even img[src]</i>
3	Jam tayang	<i>.table-theater-det tr td div</i>
4	Kategori film	<i>.table-theater-det tr td [title]</i>

#### 3.2 Analisis Pertanyaan

Beberapa pertanyaan yang nantinya akan dijadikan sebagai sampel akan dikumpulkan terlebih dahulu. Pertanyaan harus mengandung salah satu dari tiga kata tanya seperti apa, siapa, dan berapa seperti pada Tabel 4.

Tabel 4. Sampel Pertanyaan Penelitian

No	Keterangan
1	<p>Pertanyaan yang menanyakan tentang judul film yang dibintangi oleh aktor tertentu.</p> <p>Kata 'dibintangi', 'diperankan' dan 'dimainkan' memiliki makna yang sama.</p> <p>Kata 'membintangi', 'me-merankan' dan 'memainkan' memiliki makna yang sama.</p> <ul style="list-style-type: none"> <li>• Apa judul film yang dibintangi oleh X?</li> <li>• Apa saja judul film yang dibintangi oleh X?</li> <li>• Apa nama film yang dibintangi oleh X?</li> </ul>

No	Keterangan
	<ul style="list-style-type: none"> <li>• Apa saja nama film yang dibintangi oleh X?</li> <li>• Film apa yang dibintangi oleh X?</li> <li>• Film apa saja yang dibintangi oleh X?</li> <li>• Apa judul film yang dibintangi X?</li> <li>• Apa saja judul film yang dibintangi X?</li> <li>• Apa nama film yang dibintangi X?</li> <li>• Apa saja nama film yang dibintangi X?</li> <li>• Film apa yang dibintangi X?</li> <li>• Film apa saja yang dibintangi X?</li> <li>• Aktor X membintangi film apa?</li> <li>• Aktor X membintangi film apa saja?</li> </ul>
2	<p>Pertanyaan yang menanya-kan tentang aktor yang membintangi judul film tertentu.</p> <p>Kata ‘membintangi’, ‘me-merankan’ dan ‘memainkan’ memiliki makna yang sama.</p> <p>Kata ‘aktor’ dan ‘pemain’ memiliki makna yang sama</p> <ul style="list-style-type: none"> <li>• Siapa saja aktor yang membintangi film X?</li> <li>• Siapa saja yang membintangi film X?</li> <li>• Siapa yang membintangi film X?</li> <li>• Siapa aktor yang membintangi film X?</li> </ul>
3	<p>Pertanyaan yang menanya-kan tentang kategori film.</p> <p>Kata ‘layak’, ‘cocok’, ‘boleh’, dan ‘dapat’ memiliki makna yang sama</p> <ul style="list-style-type: none"> <li>• Film X layak ditonton oleh siapa?</li> <li>• Film X layak ditonton oleh siapa saja?</li> <li>• Apa kategori dari film X?</li> <li>• Apa kategori film X?</li> </ul>
4	<p>Pertanyaan yang me-nanyakan tentang durasi film</p> <ul style="list-style-type: none"> <li>• Berapa durasi dari film X?</li> <li>• Berapa durasi film X?</li> <li>• Berapa lama durasi dari film X?</li> <li>• Berapa lama durasi film X?</li> <li>• Berapa lama film X diputar?</li> </ul>
5	<p>Pertanyaan yang menanya-kan tentang sinopsis film</p> <ul style="list-style-type: none"> <li>• Apa isi sinopsis dari film X?</li> <li>• Apa isi sinopsis film X?</li> </ul>
6	<p>Pertanyaan yang menanya-kan tentang daftar film yang sedang tayang.</p> <p>Kata ‘sedang’ dan ‘lagi’ memiliki makna yang sama</p> <ul style="list-style-type: none"> <li>• Apa judul film yang sedang tayang?</li> <li>• Apa saja judul film yang sedang tayang?</li> <li>• Apa nama film yang sedang tayang?</li> <li>• Apa saja nama film yang sedang tayang?</li> <li>• Film apa yang sedang tayang?</li> <li>• Film apa saja yang sedang tayang?</li> <li>• Apa judul film yang sedang tayang?</li> <li>• Apa saja judul film yang sedang tayang?</li> <li>• Apa judul film yang sedang tayang hari ini?</li> <li>• Apa saja judul film yang sedang tayang hari ini?</li> <li>• Apa nama film yang sedang tayang hari ini?</li> <li>• Apa saja nama film yang sedang tayang hari ini?</li> <li>• Film apa yang sedang tayang hari ini?</li> <li>• Film apa saja yang sedang tayang hari ini?</li> <li>• Apa judul film yang sedang tayang hari ini?</li> <li>• Apa saja judul film yang sedang tayang hari ini?</li> <li>• Apa judul film yang hari ini sedang tayang?</li> <li>• Apa saja judul film yang hari ini sedang tayang?</li> <li>• Apa nama film yang hari ini sedang tayang?</li> <li>• Apa saja nama film yang hari ini sedang tayang?</li> <li>• Film apa yang hari ini sedang tayang?</li> <li>• Film apa saja yang hari ini sedang tayang?</li> <li>• Apa judul film yang hari ini sedang tayang?</li> <li>• Apa saja judul film yang hari ini sedang tayang?</li> <li>• Hari ini film apa yang sedang tayang?</li> <li>• Hari ini film apa saja yang sedang tayang?</li> </ul>

No	Keterangan
7	<p>Pertanyaan yang menanyakan tentang daftar film yang akan tayang</p> <ul style="list-style-type: none"> <li>• Apa judul film yang akan tayang?</li> <li>• Apa saja judul film yang akan tayang?</li> <li>• Apa judul film yang nanti akan tayang?</li> <li>• Apa saja judul film yang nanti akan tayang?</li> <li>• Apa judul film yang nanti akan tayang?</li> <li>• Apa saja judul film yang nanti akan tayang?</li> <li>• Apa nama film yang nanti akan tayang?</li> <li>• Apa saja nama film yang nanti akan tayang?</li> <li>• Film apa yang nanti akan tayang?</li> <li>• Film apa saja yang nanti akan tayang?</li> <li>• Apa judul film yang nanti akan tayang?</li> <li>• Apa saja judul film yang nanti akan tayang?</li> </ul>

### 3.3 Analisis Tokenization

Suatu kalimat akan dipotong menjadi bagian unit terkecil. Metode yang paling sederhana untuk dilakukan adalah memisahkan suatu kata berdasarkan tanda spasi. Sebagai contoh, jika pertanyaan “Siapa pemeran film X?” dilakukan pemisahan *string* berdasarkan spasi, maka hasil yang didapat adalah “Siapa”, “pemeran”, “film”, “X?”. Kelompok *string* terakhir tidak dapat memisahkan huruf X dengan simbol tanda tanya (?) karena syarat pemisahan *string* hanya berdasarkan spasi.

Untuk mengatasi masalah di atas, maka sebelum dilakukan *tokenization*, satu karakter terakhir dari suatu pertanyaan akan dihapus karena setiap kalimat pertanyaan pastinya diakhiri dengan simbol tanda tanya. Pertanyaan yang diakhiri dengan tanda tanya lebih dari satu seperti kalimat “Siapa pemeran film X???” akan menghasilkan *string* “Siapa”, “pemeran”, “film”, “X??”. Seluruh simbol tanda tanya di akhir kalimat tidak perlu dihapus karena ada beberapa judul film yang terdapat simbol tanda tanya di akhir judul tersebut. Setiap *string* yang telah dilakukan *tokenization* nantinya akan diubah menjadi huruf kecil.

### 3.4 Analisis Part of Speech

*Part of Speech* digunakan untuk mencari jenis POS pada pertanyaan dari pengguna. Kamus POS yang digunakan diambil dari situs [http://en.wiktionary.org/wiki/Category:Indonesian\\_parts\\_of\\_speech](http://en.wiktionary.org/wiki/Category:Indonesian_parts_of_speech) dan <http://kamus.sabda.org/kamus/>. Dari situs tersebut, kata tanya seperti “apa”, “siapa”, dan “berapa” termasuk dalam kategori *pronoun*. Namun kata tanya tersebut akan dimasukkan ke dalam kategori POS baru, yaitu *wh-question* (WH). Hal ini untuk memudahkan dalam mendeteksi kata tanya dari suatu *string* serta memisahkan kata tanya tersebut dengan kata *pronoun* lainnya seperti “yang”, “beberapa”, dan “kini”.

Selain POS *wh-question*, POS *verb* (VB) akan dibagi menjadi kata kerja dasar, kata kerja aktif, dan kata kerja pasif. Tujuannya untuk mengurangi ambiguitas dari suatu pertanyaan. Contohnya adalah

pertanyaan “Siapa yang memainkan film X” dengan “Siapa yang dimainkan film X” akan menghasilkan urutan POS yang sama. Untuk itu jenis POS *verb* akan dipisahkan agar proses pencocokan aturan nantinya berjalan sesuai rencana. Kata kerja aktif biasanya selalu diawali dengan kata me- seperti memainkan, memerankan, dan membintangi. Kata kerja pasif biasanya selalu diawali dengan kata di- seperti dimainkan, diperankan, dan dibintangi. Kata kerja dasar tidak menggunakan imbuhan seperti boleh, dapat, dan sebagainya. Oleh karena itu, POS *verb* terdapat 3 jenis, yaitu *verb* dasar (VB), *active verb* (VBa) dan *passive verb* (VBp). Daftar kata yang digunakan pada POS dapat dilihat pada Tabel 5.

Tabel 5. Daftar Kategori POS Yang Digunakan

POS	Keterangan POS	Kata
WH	Kata tanya	Apa, siapa, berapa
NN	Kata benda	Judul, film, nama, actor, kategori, durasi, genre, jenis, sinopsis, isi, hari, nanti
IN	Kata preposisi	Yang, oleh, dari, dalam
RB	Kata keterangan	Saja, layak, sedang, lagi, akan
VB	Kata kerja	Dapat, boleh, termasuk, tayang
VBa	Kata kerja aktif	Membintangi, memerankan, memainkan
VBp	Kata kerja pasif	Dibintangi, diperankan, dimainkan, ditonton, diputar
JJ	Kata sifat	Cocok, lama
PR	Kata ganti	Ini

*String* yang telah dilakukan *tokenization* akan disisipkan *tag* POS. Jika suatu *token* dikenali berdasarkan tabel POS pada basis data, maka *token* tersebut akan disisipkan *tag* POS yang sesuai. Tetapi *token* yang tidak dikenali akan disisipkan *tag* POS kosong. Sebagai contoh, pertanyaan “Film apa yang dimainkan Bruce Lee?” akan menghasilkan dokumen dalam bentuk JSON seperti berikut.

```
[ { "token": "film", "POS": "NN" },
  { "token": "apa", "POS": "WH" },
  { "token": "yang", "POS": "PR" },
  { "token": "dimainkan", "POS": "VBp" },
  { "token": "bruce", "POS": "" },
  { "token": "lee", "POS": "" } ]
```

### 3.5 Analisis Named Entity Recognition

NER digunakan untuk membedakan kata yang merupakan nama objek dengan yang bukan objek. Objek yang dimaksud dalam penelitian ini adalah entitas nama film dan entitas nama aktor. *String* yang akan diketahui entitasnya adalah *string* yang tidak termasuk dalam kategori POS apapun.

Namun dampak dari metode ini adalah terjadinya kegagalan dalam mendeteksi nama film atau nama aktor yang mengandung kata serapan bahasa Indonesia. Contohnya adalah nama aktor Dan Bryd,

di mana “Dan” merupakan kata penghubung yang digunakan dalam bahasa Indonesia. Sehingga hanya kata “Bryd” yang akan dicari tipe entitasnya.

Pengenalan pola pada NER bekerja berdasarkan fitur yang digunakan. Berdasarkan sampel pertanyaan yang telah dikumpulkan, fitur yang akan digunakan adalah fitur kelas kata dan fitur kontekstual. Fitur kelas kata digunakan karena hanya kata-kata yang tidak memiliki kategori POS yang akan dicari entitasnya. Fitur kontekstual dipilih karena nama entitas dapat dikenali berdasarkan kata awalan maupun kata akhiran. Sebagai contoh, nama film sering diawali dengan kata “film” dan nama aktor sering diawali dengan kata “aktor”. Fitur kontekstual tersebut dapat dilihat pada Tabel 6.

Tabel 6. Fitur Kontekstual Yang Digunakan

Nama Fitur	Keterangan	Contoh
PRE_FILM	Kata yang digunakan untuk mengawali nama film	Film
PRE_ACTOR	Kata yang digunakan untuk mengawali nama aktor	Aktor, pemeran, artis
NUMBER	Digit yang mewakili angka	1, 2, 3, dsb

Contoh pertanyaan yang terdapat entitas nama film adalah “Siapa saja aktor yang memainkan film Spiderman 2?”. Urutan proses *preprocessing*nya dan tipe entitas yang diharapkan dapat dilihat pada Tabel 7.

Tabel 7. Hasil Preprocessing Pada Pertanyaan

No.	Token	Kelas Kata	Fitur Kontekstual
1	siapa	WH	-
2	saja	JJ	-
3	aktor	NN	-
4	yang	PR	-
5	memainkan	VBa	-
6	film	NN	PRE_FILM
7	spiderman	-	-
8	2	-	NUMBER

Tahap selanjutnya adalah menyusun fitur. Proses pengenalan entitas dari *token* diperlukan informasi *token* di sekitar *token* tersebut. *Token-1* berarti *token* sebelumnya dan *token+1* berarti *token* sesudahnya. Daftar variabel dapat dilihat pada Tabel 8. Untuk aturan bahwa suatu *token* merupakan entitas film dapat dilihat pada Tabel 9 dan aturan bahwa suatu *token* merupakan entitas aktor dapat dilihat pada Tabel 10. Tabel tersebut menggambarkan beberapa aturan yang telah disusun dan minimal ada satu peraturan yang dipenuhi untuk menetapkan bahwa *token* tersebut merupakan bagian dari entitas.

Tabel 8. Daftar Variabel Yang Digunakan

Var	Fitur	Keterangan
x1	unknownPOS	Bernilai <i>true</i> jika POS dari <i>token</i> tidak diketahui.
x2	preFilm	Bernilai <i>true</i> jika fitur kontekstual dari <i>token-1</i> adalah PRE_FILM

Var	Fitur	Keterangan
x3	preActor	Bernilai <i>true</i> jika fitur kontekstual dari <i>token-1</i> adalah PRE_ACTOR
x4	activeVerbStarted	Bernilai <i>true</i> jika POS dari verb yang
		bersifat aktif (me-mainkan, memerankan, dan membintangi).
x5	passiveVerbStarted	Bernilai <i>true</i> jika POS dari <i>token-1</i> adalah <i>verb</i> yang bersifat pasif (dimainkan, diperankan, dan dibintangi).
x6	prepositionStarted	Bernilai <i>true</i> jika POS dari <i>token-1</i> adalah <i>preposition</i> (oleh, sebagai)
x7	isNum	Bernilai <i>true</i> jika fitur kontekstual dari <i>token</i> adalah NUMBER
x8	filmEntityStarted	Bernilai <i>true</i> jika <i>token-1</i> sudah dianggap sebagai <i>token</i> film
x9	actorEntityStarted	Bernilai <i>true</i> jika <i>token-1</i> sudah dianggap sebagai <i>token</i> aktor

Tabel 9. Daftar Aturan Untuk Mendeteksi Entitas Film

No	Aturan
1	x1 AND x2
2	x1 AND x4
3	x1 AND x8
4	x7 AND x8

Tabel 10. Daftar Aturan Untuk Mendeteksi Entitas Aktor

No	Aturan
1	x1 AND x3
2	x1 AND x5
3	x1 AND x6
4	x1 AND x9

Berdasarkan aturan dari Tabel 9 dan Tabel 10, maka hasil *processing* dari Tabel 9 dapat diketahui bahwa kata 'spiderman' dan angka '2' merupakan entitas film karena kata 'spiderman' memenuhi aturan dari film, yaitu x1 AND x2. Angka '2' juga memenuhi aturan dari film, yaitu x7 AND x8. Karena kedua entitas yang telah diketahui entitasnya berada di posisi berurutan, maka kedua *token* akan digabung menjadi satu. Dokumen JSON yang dihasilkan setelah proses NER adalah sebagai berikut.

```
[ { "token": "siapa", "POS": "WH" },
  { "token": "saja", "POS": "JJ" },
  { "token": "aktor", "POS": "NN" },
  { "token": "yang", "POS": "PR" },
  { "token": "memainkan", "POS": "VBa" },
  { "token": "film", "POS": "NN" },
  { "token": "spiderman 2", "POS": "", "entity": "film" } ]
```

### 3.6 Analisis Validasi Pertanyaan

Pertanyaan yang dimasukkan oleh pengguna akan dilakukan proses validasi terlebih dahulu apakah suatu pertanyaan bisa dijawab oleh sistem atau tidak. Proses validasi yang dilakukan adalah mengekstrak POS dan jenis entitas pada suatu

pertanyaan, kemudian dicocokkan dengan daftar urutan POS yang sudah disiapkan. Jika urutan POS dari pertanyaan pengguna tidak termasuk dalam daftar, maka pertanyaan tidak dianggap valid.

Daftar urutan POS atau disebut juga dengan *white list* POS akan dirancang terlebih dahulu. Cara yang dilakukan adalah mengekstrak POS dan NER pada sampel pertanyaan, kemudian daftar tersebut akan disimpan ke basis data jika *white list* POS belum terdaftar dalam tabel. Ini dilakukan agar *white list* POS yang ditampung bersifat unik sehingga tidak ada data ganda. *White list* POS yang telah dikumpulkan dapat dilihat pada Tabel 11.

Tabel 11. White List POS Hasil Ekstrak Sampel Pertanyaan

No	Aturan
1	WH NN NN IN VBp IN {entity}
2	WH RB NN NN IN VBp IN {entity}
3	NN WH IN VBp IN {entity}
4	NN WH RB IN VBp IN {entity}
5	WH NN NN IN VBp {entity}
6	WH RB NN NN IN VBp {entity}
7	NN WH IN VBp {entity}
8	NN WH RB IN VBp {entity}
9	NN {entity} VBa NN WH
10	NN {entity} VBa NN WH RB
11	WH RB NN IN VBa NN {entity}
12	WH RB IN VBa NN {entity}
13	WH IN VBa NN {entity}
14	WH NN IN VBa NN {entity}
15	NN {entity} RB VBp IN WH
16	NN {entity} RB VBp IN WH RB
17	NN {entity} JJ VBp IN WH
18	NN {entity} JJ VBp IN WH RB
19	NN {entity} VB VBp IN WH
20	NN {entity} VB VBp IN WH RB
21	WH NN IN NN {entity}
22	WH NN NN {entity}
23	WH JJ NN NN {entity}
24	WH JJ NN NN {entity}
25	WH JJ NN {entity} VBp
26	WH NN NN IN NN {entity}
27	WH NN NN NN {entity}
28	NN {entity} VB IN NN WH
29	WH NN NN IN RB VB
30	WH RB NN NN IN RB VB
31	NN WH IN RB VB
32	NN WH RB IN RB VB
33	WH NN NN IN RB VB NN PR
34	WH RB NN NN IN RB VB NN PR
35	NN WH IN RB VB NN PR
36	NN WH RB IN RB VB NN PR
37	WH NN NN IN NN PR RB VB
38	WH RB NN NN IN NN PR RB VB
39	NN WH IN NN PR RB VB
40	NN WH RB IN NN PR RB VB
41	NN PR NN WH IN RB VB
42	NN PR NN WH RB IN RB VB
43	WH NN NN IN NN RB VB
44	WH RB NN NN IN NN RB VB
45	NN WH IN NN RB VB
46	NN WH RB IN NN RB VB

### 3.7 Analisis Answer Processing

Dokumen JSON yang telah dihasilkan akan dianalisa untuk didapatkan maksud pertanyaan tersebut sehingga nantinya akan menghasilkan *query* SQL dan selanjutnya dieksekusi oleh basis data. Sebelum dianalisa, sistem perlu melakukan validasi dari dokumen JSON apakah susunan POS-nya sudah sesuai dengan kriteria atau belum. Setelah susunan POS dianggap valid, maka langkah selanjutnya adalah mengekstrak informasi yang dibutuhkan seperti kata tanya yang digunakan, kata kerja yang digunakan, dan nama entitas yang ada. Dari informasi tersebut akan dirancang *query* SQL yang selanjutnya dieksekusi oleh basis data. Hasil keluaran dari basis data akan diformat sedemikian rupa sesuai dengan jenis pertanyaan. Contohnya pertanyaan yang berhubungan dengan daftar film akan ditampilkan dalam bentuk *list*.

## 4. HASIL DAN PEMBAHASAN

Berdasarkan 260 pertanyaan yang diuji, terdapat 52 pertanyaan yang hasilnya tidak sesuai yang diharapkan. Jenis pertanyaan yang ditolak adalah aktor X membintangi film apa, judul film yang diperankan oleh aktor Zhigang Yang, dan film X dapat ditonton oleh siapa. Secara keseluruhan, sistem ini mampu memberikan menjawab dengan tepat sebesar 80%.

Tabel 12. Hasil Uji Pertanyaan

Jenis Pertanyaan	Diterima (%)	Ditolak (%)
Film yang diperankan oleh aktor tertentu	42,8	57,2
Aktor yang memainkan film tertentu	100	0
Genre suatu film	91,1	8,9
Durasi suatu film	100	0
Sinopsis suatu film	100	0
Film yang sedang tayang	100	0
Film yang akan tayang	100	0
<b>TOTAL</b>	<b>80</b>	<b>20</b>

## 5. SIMPULAN

Berdasarkan proses dan hasil penelitian yang telah dilakukan, maka dapat ditarik beberapa kesimpulan sebagai berikut.

1. *Question Answering System* dapat memahami maksud pertanyaan melalui proses *question processing* dan mencocokkan *metadata* pertanyaan dengan *metadata* yang telah disiapkan.
2. *Question Answering System* dapat memberikan jawaban yang sesuai melalui proses *answer processing* dan merangkai jawaban yang akan ditampilkan senatural mungkin.

## DAFTAR PUSTAKA

- [1] Adolphs, P., Theobald, M., Schäfer, U., Uszkoreit, H. dan Weikum, G., 2011, YAGO-QA: Answering Questions by Structured Knowledge Queries, *2011 Fifth IEEE International Conference on Semantic Computing*, California.
- [2] Ardiansyah, A., 2013, Sistem Pengenalan Entitas Dengan Perceptron Pada Tweets, *Skripsi*, Universitas Pendidikan Indonesia, Bandung.
- [3] Bhaskoro, A.T., 2013, Walker Sands Mobile Traffic Report Q3 2013: Akses Internet Via Mobile Meningkat, Diikuti Dengan Penurunan Pangsa Pasar iPad. <http://dailysocial.net/post/walker-sands-mobile-traffic-report-q3-2013-akses-internet-via-mobile-meningkat-diikuti-dengan-penurunan-pangsa-pasar-ipad>, diakses tanggal 20 November 2013.
- [4] Bird, S., Loper, E. dan Klein, E., 2009, *Natural Language Processing with Python*, O'Reilly Media, California.
- [5] Crockford, D., 2008, *JavaScript: The Good Parts*, O'Reilly Media, USA
- [6] Chung, H., Song, Y., Han, K., Yoon, D., Lee, J., Rim, H. dan Kim, S., 2004, A Practical QA System In Restricted Domain, *Proceedings of the Workshop Question Answering in Restricted Domains*, Association for Computer and Linguistics, Barcelona.
- [7] Ferrández, Ó., Izquierdo, R., Ferrández, S. dan Vicedo, J.L., 2009, Addressing Ontology-Based Question Answering with Collections of User Queries, *Information Processing & Management*, New York
- [8] Liu, B., 2011, *Web Data Mining, 2<sup>nd</sup> Edition*, Springer, Berlin
- [9] Putra, R.S., 2012, Pembangunan Sistem Lowongan Pekerjaan Dengan Dukungan Web Service, *Skripsi*, Universitas Komputer Indonesia, Bandung
- [10] Srihari, R. dan Li, W., 2000, A Question Answering System Supported by Information Extraction, *Proceedings of the Sixth Conference on Applied Natural Language Processing*, Pennsylvania
- [11] Unger, C., Bühmann, L., Lehmann, J., Ngomo, A.N., Gerber, D. dan Cimiano, P., 2012, Template-Based Question Answering over RDF Data. *Proceedings of the 21st International Conference on World Wide Web*, New York.
- [12] Watson, M., 2009, *Scripting Intelligence: Web 3.0 Information Gathering and Processing*, Apress, USA.

