EKSTRAKSI INFORMASI ARTIKEL MUSIK BERDASARKAN ONTOLOGI MENGGUNAKAN *NAMED ENTITY RECOGNITION* DAN METODE *RULE BASED*

SKRIPSI

JUWITA SHIFA RAHMAH

131402111

PROGRAM STUDI S1 TEKNOLOGI INFORMASI

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UNIVERSITAS SUMATERA UTARA

2019

EKSTRAKSI INFORMASI ARTIKEL MUSIK BERDASARKAN ONTOLOGI MENGGUNAKAN *NAMED ENTITY RECOGNITION* DAN METODE *RULE BASED*

SKRIPSI

Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah
Sarjana Teknologi Informasi

JUWITA SHIFA RAHMAH
131402111

PROGRAM STUDI S1 TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
2019

PERSETUJUAN

Judul	: EKSTRAKSI ARTIKEL MUSIK BERDASARKAN ONTOLOGI MENGGUNAKAN *NAMED ENTITY RECOGNITION* DAN METODE *RULE BASED*
Kategori	: SKRIPSI
Nama	: JUWITA SHIFA RAHMAH
Nomor Induk Mahasiswa	: 131402111
Program Studi	: SARJANA (S1) TEKNOLOGI INFORMASI
Departemen	: TEKNOLOGI INFORMASI
Fakultas	: ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
Komisi Pembimbing	:

Pembimbing 2	Pembimbing 1
Dedy Arisandi S.T, M.Kom	Dani Gunawan, S.T., M.T.
NIP. 197908312009121002	NIP. 198209152012121002

Diketahui / disetujui oleh

Program Studi S1 Teknologi Informasi

Ketua,

Romi Fadillah Rahmat, B.Comp.Sc., M.Sc.
NIP. 19860303 201012 1 004

PERNYATAAN

EKSTRAKSI INFORMASI ARTIKEL MUSIK BERDASARKAN ONTOLOGI MENGGUNAKAN *NAMED ENTITY RECOGNITION* DAN METODE *RULE BASED*

SKRIPSI

Saya mengakui bahwa skripsi adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah disebutkan sumbernya.

Medan, 25 April 2019

JUWITA SHIFA RAHMAH
131402111

UCAPAN TERIMA KASIH

Puji dan syukur penulis sampaikan kehadirat Allah SWT karena berkat rahmat dan restu-Nya penulis dapat menyelesaikan penyusunan skripsi ini sebagai syarat untuk memperoleh gelar Sarjana Komputer pada Program Studi S1 Teknologi Informasi Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.

Ucapan terima kasih penulis sampaikan kepada:

1. Bapak Dani Gunawan, ST., MT. selaku Dosen Pembimbing I dan Bapak Dedy Arisandi S.T, M.Kom selaku Dosen Pembimbing II yang telah meluangkan waktu dan pikiran untuk membimbing serta memberikan saran kepada penulis dalam menyelesaikan penelitian ini. Tanpa bantuan dan arahan dari kedua pembimbing tentu penulis tidak akan menyelesaikan penelitian ini dengan baik.
2. Bapak Prof. Dr. Opim Salim Sitompul, M.Sc selaku Dosen Pembanding I dan Bapak Ainul Hizriadi, S.Kom, M.Sc selaku Dosen Pembanding II yang telah memberikan kritik dan saran yang bermanfaat dalam penyempurnaan skripsi ini.
3. Bapak Romi Fadillah Rahmat, B.Comp.Sc., M.Sc. selaku Ketua Program Studi S1 Teknologi Informasi Universitas Sumatera Utara, Ibu Sarah Purnamawati, ST., MSc. selaku Sekretaris Program Studi S1 Teknologi Informasi Universitas Sumatera Utara dan semua dosen serta pegawai di lingkungan program studi S1 Teknologi Informasi Universitas Sumatera Utara yang telah membantu serta membimbing penulis selama proses perkuliahan.
4. Kedua orang tua penulis, Ayahanda Sarianto dan Ibunda Sulasmah yang telah membesarkan penulis dengan sabar, memberikan kasih sayang dan segala kecukupan kepada penulis, memberikan nasihat dan doa yang tiada putusnya buat penulis dalam menyelesaikan penelitian ini. Terimakasih juga kepada kakak penulis Endah Julyanmah yang selalu memberikan semangat kepada penulis.

5. Teman-teman penulis dari awal masuk kuliah, Cut Amalia, Farah Fikriyah, Maulidya Rahmah dan Ummi Kalsum Harahap yang selalu ada saat susah dan senang, senantisa memberikan bantuan dan semangat kepada penulis untuk menyelesaikan penelitian ini.
6. Putri Asyary Falam, Dita Tri Purnama Sari, Susan Rezki Ananda, Lely Khairuna, Atika Aqmarina dan Firsty Nanquita yang selalu mendengar keluh kesah penulis dan selalu bertanya kapan selesai, terimakasih telah memberikan motivasi kepada penulis.
7. Teman-teman Teknologi Informasi 2013 dan senior-senior Teknologi Informasi yang namanya tidak dapat penulis sebutkan satu persatu, terimakasih telah membantu penulis selama masa perkuliahan dan juga yang turut membantu penyelesaian skripsi ini. Terimakasih juga untuk semua pihak yang telah memberikan dukungan dan bantuan kepada penulis secara langsung maupun tidak langsung.

Semoga Allah SWT melimpahkan berkah kepada semua pihak yang telah memberikan bantuan, perhatian serta dukungan dan motivasinya kepada penulis dalam menyelesaikan skripsi ini

Medan, 25 April 2019

Penulis

ABSTRAK

Artikel yang terdapat pada suatu website sebagian besar merupakan informasi yang tidak terstruktur, hal ini mengakibatkan sulitnya mendapatkan informasi utama yang relevan dalam bentuk informasi yang terstruktur. Ekstraksi informasi merupakan suatu proses untuk mengubah informasi tidak terstruktur menjadi informasi terstruktur. Ekstraksi informasi berdasarkan ontologi bertujuan untuk menyediakan konten semantik untuk web semantik. Penelitian ini menggunakan proses *Part of Speech Tagging* untuk pelabelan kelas kata dan *Named Entity Recognition* untuk pengenalan entitas kata dalam proses *pre-processing.* Proses *pre-processing* dilakukan untuk mempermudah proses ekstraksi informasi dan penentuan *rules* dalam *rule based* untuk mendapatkan hasil ekstraksi informasi. Penelitian ini hanya berfokus kepada artikel musik yang berisi informasi tentang penyelenggaraan acara musik dan mengambil 4 jenis entitas yaitu nama acara, nama artis, waktu dan tempat.

Kata Kunci : Ekstraksi informasi, Ontologi, *Part of Speech Tagging*, *Named Entity Recognition*, *rule based*, artikel musik, acara musik

INFORMATION EXTRACTION OF MUSIC ARTICLE BASED ON ONTOLOGY USING NAMED ENTITY RECOGNITION AND RULE BASED METHOD

ABSTRACT

Articles contained on a website are mostly unstructured information, this results in the difficulty of obtaining relevant information in the form of structured information. Information extraction is a process of converting unstructured information to structured information. Information extraction based on ontology aims to provide semantic content for the semantic web. This research uses Part of Speech Tagging (POS Tagging) for labeling the word class and Named Entity Recognition (NER) for identification of word entities in the pre-processing process. The pre-processing process is used to simplify the information extraction process and determine the rules in the rule-based method to obtain information extraction results. This research focuses only on music articles that contain information about music event and takes 4 types of entities that are event, artist, time and place.

Keyword : Information Extraction, Ontology, Part of Speech Tagging, Named Entity Recognition, rule based, music article, music event

DAFTAR ISI

	Hal.
Persetujuan	i.
Pernyataan	ii.
Ucapan Terima Kasih	iii.
Abstrak	v.
Abstract	vi.
Daftar Isi	vii.
Daftar Tabel	ix.
Daftar Gambar	x.
Bab 1 Pendahuluan	
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan Penelitian	3
1.4. Batasan Masalah	3
1.5. Manfaat Penelitian	3
1.6. Metodologi Penelitian	3
1.7. Sistematika Penulisan	4
Bab 2 Landasan Teori	
2.1. *Text Mining*	6
2.2. *Information Extraction*	6
2.3. Tokenisasi	7
2.4. *Part of Speech Tagging (POS Tagging)*	7
2.4.1. *Hidden Markov Model* (HMM)	9
2.4.2. *Algoritma Viterbi*	9
2.5. *Named Entity Recognition* (NER)	10
2.6. *Rule Based*	11
2.7. *Ontology*	11

2.7.1. *Music Ontology* 12
2.8. *Resource Data Framework* (RDF) 14
2.9. *SPARQL* 15
2.10. *Apache Jena Fuseki* 16
2.11. Evaluasi Kinerja Sistem 16
2.12. Penelitian Terdahulu 17

Bab 3 Analisis dan Perancangan Sistem
3.1. Data yang Digunakan 20
3.2. Analisis Sistem 20
3.2.1. *Input* 22
3.2.2. *Pre-processing* 22
3.2.3. *Process* 23
3.2.4. *Database Server* 30
3.2.5. *Output* 30
3.3. Perancangan Sistem
3.3.1. Rancangan Tampilan Halaman Utama 31
3.3.2. Rancangan Tampilan Halaman Proses 32
3.3.3. Rancangan Tampilan Halaman RDF 33

Bab 4 Implementasi dan Pengujian
4.1. Implementasi Sistem 35
4.1.1. Spesifikasi Perangkat Keras 35
4.1.2. Spesifikasi Perangkat Lunak 35
4.1.3. Implementasi Perancangan Antarmuka 36
4.2. Pembangunan Server 45
4.3. Pengujian Sistem 45

Bab 5 Kesimpulan dan Saran
5.1. Kesimpulan 50
5.2. Saran 51

DAFTAR PUSTAKA 52

DAFTAR TABEL

		Hal.
Tabel 2.1.	Tabel Tagset untuk Bahasa Indonesia	8
Tabel 2.2.	Tag Entitas	10
Tabel 2.3.	Contoh Format Data Entitas	11
Tabel 2.4.	Penelitian Terdahulu	19
Tabel 3.1.	Kata yang Diperiksa Sebelum Tag Time	26
Tabel 3.2.	Model Database file RDF dalam bentuk N-Triples	30
Tabel 4.1.	Hasil Ekstraksi Informasi Artikel Musik	48
Tabel 4.2.	Nilai *F-Score* Hasil Pengujian Sistem	49

DAFTAR GAMBAR

		Hal.
Gambar 2.1.	Contoh Input Teks	7
Gambar 2.2.	Hasil Proses Tokenisasi	7
Gambar 2.3.	Graph Daftar Instilah Dalam Ontologi Musik	13
Gambar 2.4.	RDF Graph dengan 2 node (subjek dan objek) dan sebuah triple yang menghubungkannya (predikat)	14
Gambar 2.5.	RDF triple (dalam bentuk graph)	15
Gambar 2.6.	Contoh *SPARQL Query*	16
Gambar 3.1.	Arsitektur Umum	21
Gambar 3.2.	Contoh Hasil *Tokenizing*	22
Gambar 3.3.	Contoh Hasil *POS Tagging*	23
Gambar 3.4.	Contoh Hasil *Named Entity Recognition*	23
Gambar 3.5.	Perancangan Proses Pengambilan Entitas Nama	24
Gambar 3.6	Perancangan Proses Pengambilan Entitas Title / Nama Acara	25
Gambar 3.7.	Perancangan Proses Pengambilan Entitas Waktu	26
Gambar 3.8.	Perancangan Proses Pengambilan Entitas Tempat	27
Gambar 3.9.	Graph dan Relasi Konsep Ontologi Musik pada Penelitian ini	28
Gambar 3.10.	Contoh RDF Graph Dari Hasil Ektraksi Informasi	28
Gambar 3.11.	Contoh Hasil Ekstraksi Informasi Dalam Bentuk RDF Berdasarkan Ontologi Musik	29
Gambar 3.12.	Rancangan Tampilan Halaman Utama	31
Gambar 3.13.	Rancangan Tampilan Halaman Proses	32
Gambar 3.14.	Rancangan Tampilan Halaman RDF	33
Gambar 4.1.	Tampilan Halaman Utama	36
Gambar 4.2.	Tampilan Halaman Proses	37
Gambar 4.3.	Tampilan Hasil *POS Tagging*	37
Gambar 4.4.	Tampilan Hasil *Named Entity Recognition*	38
Gambar 4.5.	Tampilan Hasil Ekstraksi yang ditampilkan dalam Bentuk RDF	43

Gambar 4.6. Tampilan Halaman Pemanggilan Output File RDF 44
Gambar 4.7. Tampilan Hasil Dari Pemanggilan File RDF 45
Gambar 4.8. Tampilan pada *Server Apache Jena Fuseki* File RDF 45

BAB I

PENDAHULUAN

1.1. Latar Belakang

Sekarang ini banyak sekali informasi yang dapat di akses di internet. Salah satunya adalah informasi yang berkaitan dengan musik. Informasi atau artikel musik yang terdapat di internet berbagai macam jenisnya, contohnya yaitu artikel yang membahas tentang sebuah perilisan musik oleh seorang artis ataupun penyelenggaraan acara musik atau disebut juga konser musik. Meskipun tujuan artikel-artikel tersebut adalah untuk menyampaikan informasi tetapi sebagian besar data atau informasi yang ada di internet ditampilkan secara tidak terstruktur melalui website. Salah satu cara untuk mendapatkan informasi terstruktur yaitu melalui ekstraksi informasi. Ekstraksi informasi bisa diartikan sebagai sebuah proses untuk menemukan informasi terstruktur dari dokumen tidak terstruktur atau semi terstruktur yang merupakan salah satu bagian dari Pemrosesan Bahasa Alami (Jiang, 2012). Secara garis besar proses ekstraksi informasi terdiri dari dua tahap, yaitu mengidentifikasi informasi yang relevan di dalam sebuah dokumen teks, kemudian menyimpannya ke dalam bentuk terstruktur untuk digunakan kemudian.

Sebelumnya terdapat beberapa penelitian ekstraksi informasi telah dilakukan dengan menggunakan beberapa metode. Ismaya (2014) menggunakan *rule-based classification*, dan metode *preprocessing* yang digunakan adalah *POS Tagging* untuk melakukan ekstraksi informasi dari sekumpulan dokumen Laporan Hasil Pemeriksaan (LHP) Badan Pemeriksa Keuangan (BPK) RI. Akurasi dari data latih dan data uji masing-masing adalah 98,27% dan 89,77%. Penurunan akurasi antara lain disebabkan oleh penggunaan aplikasi pdf2text yang tidak memberikan hasil konversi yang identik atara data input dan output, serta penggunaan metode wordmatch untuk klasifikasi.

Permadi (2009) menggunakan *Partial Tree Alignment* (DEPTA). Dengan metode ini halaman web akan dimodelkan dengan struktur tree sebelum akhirnya di

ekstraksi. Sistem DEPTA menghasilkan recall 97%, precision 37%, the harmonic mean 50% dan the e-measure 61%. Namun sistem ini hanya dapat berjalan baik pada bentuk website yang mempunyai banyak blok tabel, misalnya pada website komersial. Sebaliknya jika pada bentuk website berita, sistem ini tidak akan berjalan dengan baik.

Penelitian yang dilakukan oleh Susanti (2015) dalam ekstraksi informasi konten web menggunakan pendekatan berbasis ontologi yang disebut *Ontology-Based Information Extraction* (OBIE). Studi kasus dalam penerapan ekstraksi informasi ini dilakukan menggunakan dataset "Lonely Planet". Dari hasil evaluasi pengujian menunjukkan bahwa hasil *Recall* 62%, *Precision* 73% dan *F-measure* 67%.

Riaddy (2016) menggunakan pendekatan *Supervised Learning* dalam melakukan ekstraksi informasi pada makalah ilmiah dengan menggabungkan tiga algoritma *supervised learning* yaitu *Support Vector Machine* (SVM), *Naïve Bayes*, dan *Random Forest.* Implementasi dari ketiga *classifier* tersebut pada penelitian ini menggunakan *library* dari Wakaito Environment for Knowledge Analysis (WEKA). Informasi yang diekstraksi dari makalah ilmiah pada penelitian ini yaitu judul, nama penulis, afiliasi, alamat, *email* dan *website*. Nilai akurasi kata data training dan data testing dari tiap informasi yang diekstraksi dari makalah ilmiah yaitu diatas 90%.

Maka pada penelitian ini penulis melakukan ekstraksi informasi artikel musik yang berkaitan dengan penyelenggaraan acara musik atau konser menggunakan *NER (Named Entity Recognition)* dalam proses *pre-processing* dan metode *rule based* yang hasil akhirnya diterapkan berdasarkan ontologi musik.

1.2. Rumusan Masalah

Artikel-artikel yang dibuat pada suatu website sebagian besar merupakan informasi yang tidak terstruktur, hal tersebut berakibat sulitnya mendapatkan informasi utama yang relevan dalam bentuk informasi yang terstruktur. Berdasarkan hal ini, maka diperlukan sebuah pendekatan untuk membantu melakukan ekstraksi informasi sehingga mendapatkan informasi yang terstruktur.

1.3. Tujuan Penelitian

Tujuan utama penelitian ini yaitu mendapatkan data yang terstruktur dari artikel web dengan melakukan ekstraksi informasi yang hasilnya di kategorikan berdasarkan

ontologi musik yang telah ada dengan menggunakan *Named Entity Recognition* dan metode *rule based*.

1.4. Batasan Masalah

Sistem ekstraksi informasi yang dikembangkan memiliki batasan atau ruang lingkup yang mencangkup :

1. Sistem hanya dapat mengekstraksi artikel musik yang berkaitan dengan pelaksaan konser.
2. Sistem hanya mengambil data dari 4 jenis *entity* yaitu nama acara, artis, tempat dan waktu.
3. Sistem hanya bekerja dengan baik apabila penulisan kata, tanda baca dan penggunaan huruf kapital sudah sesuai dengan Ejaan Bahasa Indonesia.

1.5. Manfaat Penelitian

Manfaat yang diperoleh dari penelitian ini adalah :

1. Membantu memberikan informasi yang relevan dan terstruktur.
2. Hasil dapat digunakan sebagai data dalam pembuatan web semantik dan sistem *question answering*.
3. Dapat menjadi acuan dalam penelitian selanjutnya terkait penggunaan ontologi.

1.6. Metodologi Penelitian

Tahapan-tahapan yang dilakukan selama penelitian adalah sebagai berikut:

1. Studi Literatur

   Tahap ini dilaksanakan untuk mengumpulkan dan mempelajari informasi-informasi yang diperoleh dari buku, jurnal, dan berbagai sumber referensi lain yang berkaitan dengan penelitian tentang ekstraksi informasi. Topik-topik berkaitan dengan berbagai pendekatan dalam ekstraksi informasi dipelajari sebagai dasar pengetahuan dalam penyelesaian permasalahan penelitian ini.
2. Analisis Permasalahan

Pada tahap ini dilakukan analisis terhadap berbagai informasi yang telah diperoleh dari berbagai sumber yang terkait dengan penelitian agar didapatkan metode yang tepat untuk menyelesaikan masalah dalam penelitian ini.

3. Perancangan dan Implementasi Sistem

   Tahap ini dilakukan perancangan sistem mulai dari perancangan arsitektur, pengumpulan data dan perancangan *user interface* (UI) untuk menyelesaikan permasalahan yang terdapat di dalam tahap analisis. Kemudian dilanjutkan dengan mengimplementasikan hasil analisis dan perancangan ke dalam sistem.

4. Implementasi

   Pada tahap ini dilakukan proses implementasi kode program sistem aplikasi menggunakan bahasa pemrograman yang telah dipilih sesuai dengan analisis dan perancangan yang telah dilakukan.

5. Pengujian

   Pada tahap ini dilakukan pengujian terhadap hasil sistem sesuai dengan kebutuhan yang telah ditentukan sebelumnya serta memastikan sistem berjalan sesuai dengan yang diharapkan.

6. Dokumentasi dan Penyusunan Laporan

   Pada tahap ini dilakukan dokumentasi dan penyusunan laporan hasil evaluasi dan analisis ekstraksi informasi artikel musik yang dilakukan dengan menggunakan *named entity recognition* dan metode *rule based*.

1.7. Sistematika Penulisan

Sistematika penulisan dari skripsi ini terdiri atas lima bagian utama sebagai berikut:

Bab 1: Pendahuluan

Bab ini berisi latar belakang dari penelitian yang dilaksanakan, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, metodologi penelitian, serta sistematika penulisan dari penyusunan skripsi ini.

Bab 2: Landasan Teori

Bab ini berisi teori-teori yang diperlukan untuk memahami permasalahan yang dibahas pada penelitian ini. Teori-teori yang berhubungan dengan *text mining*, *text pre-*

processing, Information Extraction, Ontology, POS Tagging, Named Entity Recognition, HMM (Hidden Markov Model) akan dibahas pada bab ini.

Bab 3: Analisis dan Perancangan

Bab ini berisi tentang tahap-tahap dalam menganalisis permasalahan dan kebutuhan sistem, serta perancangan sistem dan alur proses kerja sistem.

Bab 4: Implementasi dan Pengujian

Bab ini berisi pembahasan tentang implementasi dari perancangan yang telah dijabarkan pada bab 3. Selain itu, hasil yang didapatkan selama proses yang terjadi pada penelitian juga dijabarkan pada bab ini.

Bab 5: Kesimpulan dan Saran

Bab ini berisi ringkasan serta kesimpulan dari rancangan yang telah dibahas pada bab-bab sebelumnya, serta berisi saran-saran yang diajukan untuk pengembangan penelitian selanjutnya.

BAB II

LANDASAN TEORI

2.1. *Text Mining*

Teks mining didefinisikan sebagai proses menemukan *pattern*, model, petunjuk, *trends*, atau *rules* yang berguna atau menarik dari teks tidak terstruktur. Beberapa teknik telah diusulkan dalam *text mining* termasuk *information extraction*, *information retrieval*, *natural language processing*, *categorization*, *clustering etc.* (Mooney, 2002). *Text mining* sebenarnya tidak jauh berbeda dengan *data mining*, yang membedakan hanyalah sumber data yang digunakan (Charisma, 2016). Pada *data mining* data yang digunakan adalah data yang terstruktur, sedangkan pada *text mining* data yang digunakan adalah data yang tidak terstruktur berupa teks (Charisma, 2016). Tujuan dari *text mining* secara keseluruhan adalah pada dasarnya untuk mengubah suatu teks menjadi suatu data yang dapat dianalisis (Charisma, 2016).

2.2. *Information Extraction*

Information Extraction bisa diartikan sebagai sebuah proses untuk menemukan informasi terstruktur dari dokumen tidak terstruktur atau semi terstruktur (Jiang, 2012). Ekstraksi informasi merupakan salah satu bagian dari Pemrosesan Bahasa Alami yang di dedikasikan pada masalah umum dalam mendeteksi entitas yang mengacu pada teks bahasa alami, relasi yang terdapat di dalamnya dan peristiwa-peristiwa yang diikutinya. Secara tidak langsung tujuannya adalah mendeteksi elemen-elemen seperti “siapa” melakukan “apa” ke “siapa” , “kapan” dan “dimana” (Marquez *et al*. 2008). Ekstraksi informasi termasuk tugas penting dalam *text mining* dan telah dipelajari secara ekstensif dalam berbagai komunitas penelitian termasuk *natural languange processing*, *information retrieval*, dan *Web mining* (Jiang, 2012).

Proses Ekstraksi Informasi bisa melakukan dua teknik pendekatan yaitu teknik berdasarkan aturan dan berdasarkan statistik, teknik berdasarkan aturan dibuat dan dirancang oleh pakar, teknik berdasarkan statistik dibangun dan dibuat dengan menggunakan data latih, data latih sesuai dengan teks yang dijadikan sebagai objek Ekstraksi Informasi (Ilyas & Khodra, 2015). Pendekatan yang paling umum dalam melakukan ekstraksi informasi adalah dengan penggalian teks/*text mining.* Penggalian teks didefinisikan sebagai cara melakukan penemuan otomatis informasi yang tidak ditemukan sebelumnya, dilakukan oleh komputer secara otomatis dan diambil dari sumber informasi yang berbeda (Ilyas & Khodra, 2015)

2.3. Tokenisasi

Proses *Tokenization* atau tokenisasi yaitu proses memecah teks yang berupa kalimat atau paragraf menjadi token-token (kata-kata). Pada proses ini setiap token akan dipisahkan dengan spasi sebagai contoh yaitu kalimat pada gambar 2.1.

Pada hari Selasa saya pergi ke kampus untuk masuk kelas Pemrograman Web, kemudian setelah kelas selesai saya pulang untuk mengerjakan tugas.

Gambar 2.1 Contoh Input Teks

Contoh kalimat input pada gambar 2.1 yang telah dilakukan proses tokenisasi akan menghasilkan kumpulan kata seperti gambar 2.2.

“Pada”,“hari”,”Selasa”,”saya”,”pergi”,”ke”,”kampus”,”untuk”,”masuk”,”kelas”, ”Pemrograman”,”Web”,”kemudian”,”setelah”,”kelas”,”selesai”,”saya”,”pulang”, ”untuk”,”mengerjakan”,”tugas”

Gambar 2.2 Hasil Proses Tokenisasi

2.4. *Part of Speech Tagging* (*POS Tagging*)

POS Tagging yang juga dikenal dengan istilah *grammatical tagging* adalah proses memberikan informasi mengenai kelas kata secara otomatis kepada sebuah kata dalam kalimat (Jurafsky, 2006). Secara umum kelas kata dapat dibedakan menjadi kelas Kata Benda (*Noun*), Kata Kerja (Verb), Kata Sifat (Adjective), Kata Keterangan (Adverb), Kata Penghubung (Conjuction Word), Kata Ganti Orang (Personal Pronoun), Kata

Bilangan (Numeral), dan Lainnya (Other) termasuk didalamnya berupa tanda baca. *POS Tagging* dapat dilakukan dengan beberapa cara yaitu dengan pendekatan berbasis aturan (*rule* based), probabilistik, dan pendekatan transformasional (Wicaksono & Purwarianti, 2010). Pada *POS tagging*, urutan *tag* tidak dapat diamati secara langsung. Pengamatan secara langsung hanya dapat dilakukan terhadap urutan kata. Dari urutan kata tersebut harus dicari urutan *tag* yang paling tepat. Setiap kata yang dikenali dari proses *POS Tagging* akan digunakan dalam klasifikasi.

Infomasi kelas kata yang diperoleh dari proses *part of speech tagging* sangat membantu dalam berbagai aplikasi komputasional linguistik yang ada, seperti membantu dalam *word sense disambiguation*, meningkatkan akurasi dalam pelafalan kata (*text to speech*), membantu proses *stemming* dalam *Information Retrieval*, membantu dalam linguistik forensik dan berbagai aplikasi-aplikasi lainnya.

Contoh *POS Tagging*:

Kakak(NN); pergi(VB); liburan(NN); ke(IN); Jakarta(NNP);

Tabel 2.1. Tabel Tagset untuk Bahasa Indonesia
(Wicaksono & Purwarianti, 2010)

No	Tag	Keterangan	Contoh
1	OP	*Open Parenthesis*	({[
2	CP	*Close Parenthesis*	)}]
3	GM	*Slash*	/
4	;	*Semicolon*	;
5	:	*Colon*	:
6	“	*Quotation*	“
7	.	*Sentence Terminator*	. ! ?
8	,	*Comma*	,
9	-	*Dash*	-
10	...	*Ellipsis*	...
11	JJ	*Adjective*	Kaya, senang, bersih.
12	RB	*Adverb*	Sangat, hanya, justru, nanti.
13	NN	*Common Noun*	Monyet, bawah, sekarang, rupiah.
14	NNP	*Proper Noun*	Budiono, Indonesia, Jakarta.
15	NNG	*Genetive Noun*	Selanjutnya, orangnya, bukunya.
16	VBI	*Intransitive Verb*	Kerja, pergi, tumbuh.
17	VBT	*Transitive Verb*	Membeli, membuat
18	IN	*Preposition*	Di, ke, dari, dengan.
19	MD	*Modal*	Bisa, boleh, harus.

20	CC	*Coor-Conjuction*	Dan, tetapi, atau.
21	SC	*Subor-Conjuction*	Jika, ketika.
22	DT	*Determiner*	Ini, itu, setiap
23	UH	*Interjection*	Ya, aduh, wah
24	CDO	*Ordinal Numerals*	Pertama, kedua
25	CDC	*Collective Numerals*	Bertiga
26	CDP	*Primary Numerals*	Satu, dua
27	CDI	*Irregular Numerals*	Beberapa
28	PRP	*Personal Pronouns*	Saya, kamu
29	WP	*WH-Pronouns*	Apa, siapa
30	PRN	*Number Pronouns*	Kedua-duanya
31	PRL	*Locative Pronouns*	Sini, situ, sana
32	NEG	*Negation*	Bukan, tidak
33	SYM	*Symbol*	@#$%^&
34	RP	*Particles*	Pun, kah
35	FW	*Foreign Words*	Foreign, word

*2.4.1. Hidden Markov Model (*HMM)

Hidden Markov Model (HMM) adalah sebuah model statistik dari sebuah sistem yang melakukan perhitungan probabilitas dari suatu kejadian yang tidak dapat diamati berdasarkan kejadian yang dapat diamati (Jurafsky, 2000). Perhitungan probabilitas dilakukan dengan melihat kejadian-kejadian lain yang dapat diamati secara langsung.

Dalam HMM, bagian yang dapat diamati disebut *observed state* sedangkan bagian yang tersembunyi disebut *hidden state* (Wibisono, 2008). HMM memungkinkan pemodelan sistem yang mengandung *observed state* dan *hidden state* saling terkait. Pada kasus *POS tagging*, urutan kelas kata tidak dapat diamati secara langsung sehingga dijadikan *hidden state* dan yang menjadi *observed state* adalah urutan kata. Selain itu, HMM dapat digunakan dalam pengenalan suara, *parsing*/*chunking*, ekstraksi informasi, dan peringkasan teks (Wibisono, 2008).

Ciri-ciri HMM adalah :

- Observasi diketahui tetapi urutan state tidak diketahui
- Observasi adalah fungsi probabilitas keadaan
- Perpindahan antar state adalah dalam bentuk probabilitas (Evans, 1999).

2.4.2. Algoritma Viterbi

Algoritma viterbi merupakan sebuah algoritma *dynamic programming* untuk menemukan barisan *hidden state* yang paling maksimal dari suatu barisan observasi

(Rabiner, 1999). Barisan *hidden state* yang dicari harus merupakan barisan yang optimal untuk bisa memodelkan barisan observasi dari suatu model. Algoritma viterbi digunakan untuk menemukan kemungkinan rangkaian status yang tersembunyi (biasa disebut Viterbi path) yang dihasilkan pada rangkaian pengamatan kejadian, terutama dalam lingkup HMM. Ide utama algoritma viterbi adalah mengingat sekuens untuk setiap posisi tertentu (setiap iterasi, setiap panjang kalimat). Apabila kita telah sampai pada kata terakhir, maka akan dilakukan *backtrace* untuk mendapatkan sekuens terbaik (Putra, 2018).

2.5. *Named Entity Recognition* (NER)

Named Entity Recognition atau pengenalan entitas bernama merupakan bagian penting dari proses ekstraksi informasi, yang berguna untuk mengidentifikasi jenis kata (entitas kata) dari setiap kata yang ada dalam sebuah teks (Setiyoaji, 2017). NER dapat diselesaikan dengan pelabelan urutan kata statistik (*statistical sequence-labelling)* yang mendeteksi batas atau segmen dan tipe dari *named-entity*. NER dapat digunakan untuk mengidentifikasi dan mendeteksi entitas seperti nama orang, lokasi, organisasi, waktu, kuantitas atau bilangan, dan lain lain. Proses NER dapat berjalan dengan baik dan benar apabila teks yang diperiksa sudah memiliki *case* yang benar.

NER merupakan komponen pokok pada banyak aplikasi NLP (*Natural Language Processing*) seperti sistem *question answering*, *information extraction*, *clustering*, pendeteksian topik.

Tabel 2.2 Tag Entitas (Fachri, 2014)

Entitas	Tag Entitas	Keterangan
Person	PERSON	Nama orang
Location	LOCATION	Nama tempat atau lokasi
Organization	ORGANIZATION	Nama organisasi
Time	TIME	Keterangan waktu atau tanggal
Quantity	QUANTITY	Satuan jumlah
Other	OTHER	Lain-lain

Tabel 2.3 Contoh Format Data Entitas (Fachri, 2014)

Entitas	Contoh
Person	<PERSON>Ardie Bakrie</PERSON>
Location	<LOCATION>Medan<LOCATION>
Organization	<ORGANIZATION>Golkar</ORGANIZATION>
Time	<TIME>Juli</TIME>
Quantity	<QUANTITY>10 orang</QUANTITY>
Other	<OTHER>Makan</OTHER>

2.6. *Rule Based*

Rule based merupakan pengetahuan yang disimpan dalam bentuk peraturan (Hayadi, 2016). *Rule based* ini digunakan untuk menyimpan pengetahuan dalam suatu bentuk berupa *rule* yang memiliki peran sebagai pemecah masalah dalam menyelesaikan suatu tugas. *Rule* yang digunakan dalam *rule base* sendiri dibuat berdasarkan pengetahuan yang didapat oleh manusia berdasarkan masalah yang ada kemudian diubah ke dalam bahasa yang dapat dimengerti komputer. *Rule base* telah banyak juga digunakan pada beberapa aplikasi di bidang medis, ekonomi, dan lain-lain.

2.7. *Ontology*

Ontologi adalah sebuah deskripsi formal tentang sebuah konsep secara eksplisit dalam sebuah domain, properti dari setiap konsep beserta dengan batasannya (Noy & McGuinnes, 2001). Ontologi merupakan suatu konsep pada pemrosesan informasi semantik dimana informasi semantik tersebut merepresentasikan relasi semantik, seperti kelas dan relasi, yang memberikan makna terhadap kosakata (Horridge, 2011). Secara teknis, ontologi direpresentasikan dalam bentuk class, property, facet, dan instances. Class menerangkan konsep atau makna dari suatu domain. Class adalah kumpulan dari elemen dengan sifat yang sama. Sebuah class bisa memiliki sub class yang menerangkan konsep yang lebih spesifik.

Property merepresentasikan hubungan di antara dua individu. Property menghubungkan individu dari domain tertentu dengan individu dari range tertentu.

Facet digunakan untuk merepresentasikan informasi atau batasan tentang property. Ada dua jenis facet, yaitu cardinality dan value type. Cardinality facet merepresentasikan nilai eksak yang bisa digunakan untuk slot pada suatu kelas tertentu. Cardinality facet dapat bernilai single dan multiple cardinality. Value type menggambarkan tipe nilai yang dapat memenuhi property, seperti string, number, boolean, dan enumerated.

2.7.1 Music Ontology

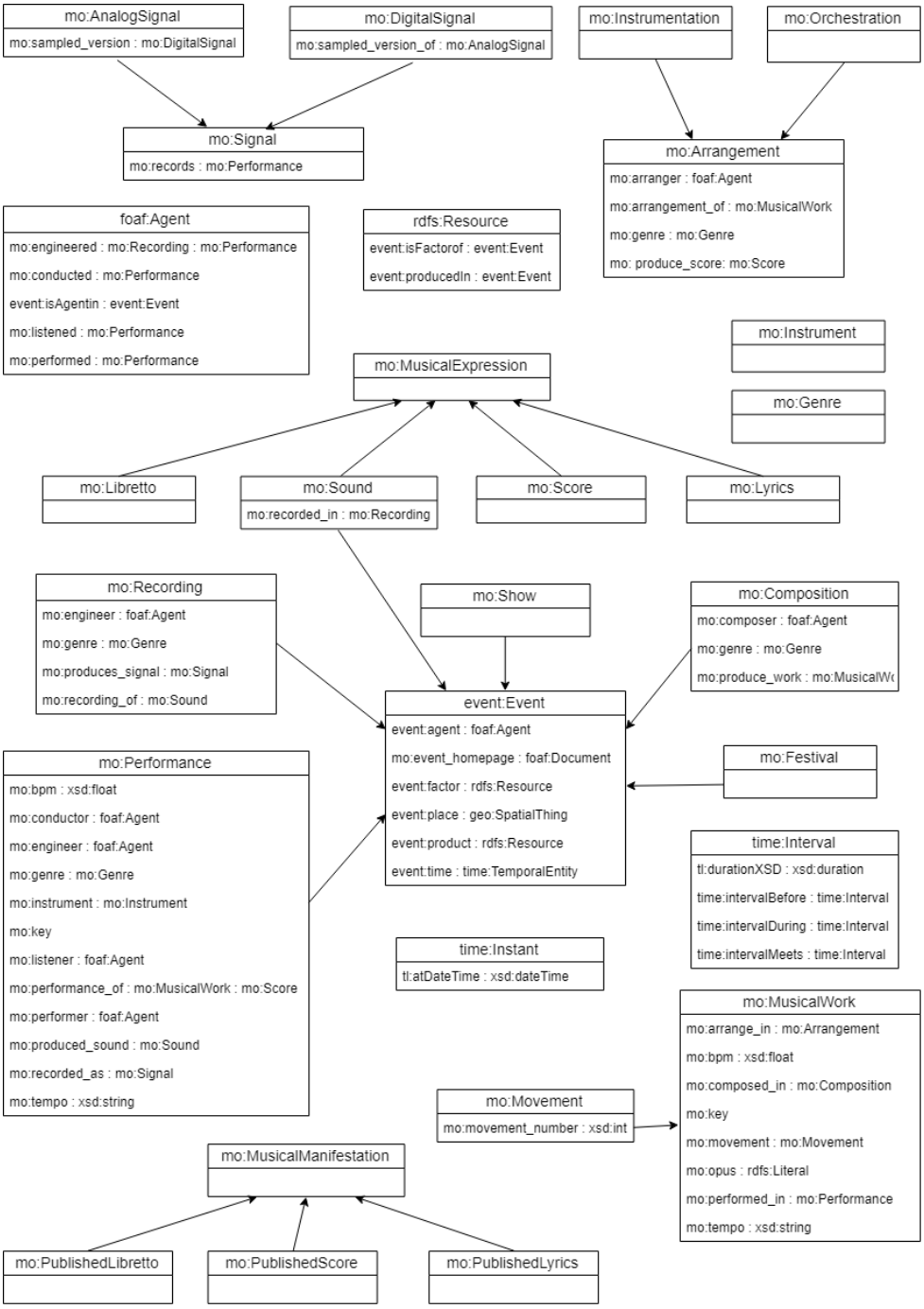
Ontologi musik menyediakan kosakata yang digunakan untuk publikasi dan menghubungkan data yang berhubungan dengan musik pada web dalam skala besar. Data dari ontologi musik dapat dipublikasi kan oleh siapapun sebagai bagian dari website atau API dan dihubungkan dengan data yang telah ada, dengan tujuan untuk membuat web berisi data yang berhubungan dengan musik.

Ontologi musik menyediakan framework yang berguna untuk membangun sebuah website musik yang juga adalah API, membuka data yang berhubungan dengan musik sembari memastikan data tersebut dapat digunakan bersama-sama dengan data dari sumber lain, mengintegrasikan data yang berkaitan dengan musik dari berbagai sumber, memperkaya hasil pencarian pada *search-engine* seputar lagu, artis, karya musik dll dan juga merancang sebuah skema *database* atau model domain yang berkaitan dengan musik.

Ontologi musik dibangun pada RDF, sebuah teknologi yang dikembangkan oleh W3C. RDF membolehkan data untuk di deskripsikan sebagai "*triples*": *subject*, *predicate* dan *object* contohnya "lagu ini" "adalah bagian dari" "album ini". Ontologi musik ditentukan menggunakan OWL, yang menyediakan seperangkat konstruksi untuk mendeksripsikan model domain pada RDF.

Ontologi musik sebagian besar dipengaruhi oleh beberapa ontologi yaitu:

1. **FOAF,** kosakata untuk mendeksripsikan orang, sekelompok orang, dan organisasi.
2. **Event Ontology,** kosakata untuk mendeskripsikan *event* atau acara
3. **Timeline Ontology,** kosakata untuk mendeskripsikan interval waktu dan jenak pada beberapa *timeline* (yang mungkin berhubungan) contoh garis waktu pada audio.
4. **FRBR Ontology,** kosakata untuk mendeskripsikan pekerjaan, ekspresi, manifestasi dan barang, dan hubungan mereka.



Gambar 2.3. Graph Daftar Istilah Dalam Ontologi Musik

Gambar 2.3. menunjukkan graph dari konsep ontologi musik yang menunjukkan daftar istilah-istilah yang dipakai dan hubungan antara satu dan lainnya dalam musik ontologi untuk menjelaskan informasi tentang alur kerja dalam sebuah penciptaan sebuah karya musik atau alur kerja dalam penciptaan sesuatu yang berhubungan dengan musik (*Composition*, *Performance*, *Arrangement*, *Recording*, dsb.).

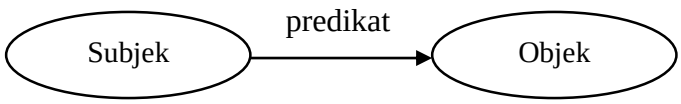
2.8. *Resource Data Framework* (RDF)

Resource Description Framework (RDF) merupakan *framework* untuk merepresentasikan informasi dari Web. Dokumen RDF ditulis dalam XML. Bahasa yang digunakan oleh RDF disebut RDF / XML. Dengan menggunakan XML, informasi RDF dengan mudah dapat dipertukarkan antara berbagai jenis komputer menggunakan berbagai jenis sistem operasi dan bahasa aplikasi.

Informasi direpresentasikan dalam bentuk *triples* yang berisi subjek, predikat dan objek dimana:

a) Subjek merupakan pengenal (*identifier*) pada baris.
b) Predikat merupakan pengenal (*identifier*) pada kolom, dan
c) Objek merupakan nilai atau *value* dari sel tersebut.

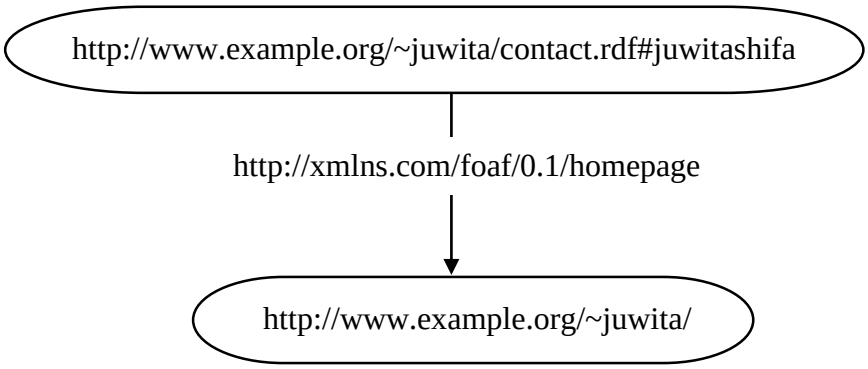
Satu set *triplets* apabila direpresentasikan dalam bentu RDF *graph* digambarkan sebagai sebuah node dan panah, dimana didalamnya setiap *triple* direpresentasikan sebagai hubungan antara node – panah – node.



Gambar 2.4. RDF graph dengan 2 node (subjek dan objek) dan sebuah triple yang menghubungkannya (predikat)

Subjek dalam RDF bisa dalam bentuk URI atau *blank node*, Predikat dalam bentuk URI dan objek dalam bentuk URI, *blank node* atau *literal*. Contoh dari triple seperti ditunjukkan dalam Gambar 2.5. Gambar tersebut menunjukkan "Juwita Shifa *has homepage* http://www.example.org/~juwita" semua elemen pada triple ini adalah *resource* yang ditetapkan dengan URI. *Resource* pertama yaitu http://www.example.org/~juwita/contact.rdf#juwitashifa (subjek) yang dimaksudkan untuk mengidentifikasi Juwita Shifa. Dapat diperhatikan bahwa *resource* tersebut tepatnya mendeskripsikan bagaimana untuk mendapatkan sebuah dokumen RDF dan serta bagaimana cara mendapatkan node RDF juwitashifa di dalamnya. *Resource* kedua yaitu http://xmlns.com/foaf/0.1/homepage (predikat) merupakan predikat *homepage*

dari kosa kata dalam FOAF (*Friend-of-a-friend*). *Resource* yang terakhir (objek) adalah *homepage* Juwita http://www.example.org/~juwita/.



Gambar 2.5. RDF triple (dalam bentuk graph)

2.9. *SPARQL*

SPARQL atau *SPARQL Protocol and RDF Query Language* merupakan bahasa *query* terstruktur pada *web* yang dapat diakses dalam format *RDF*. Selain itu, *SPARQL* merupakan protokol akses data yang distandarisasi oleh *RDF Data Access Working Group (DAWG) of the World Wide Web Consortium* untuk *semantic web*. *SPARQL* memiliki perbedaan dengan *SQL*, yakni *SPARQL* melakukan pemrosesan *query* dengan mengakses data web yang terhubung melalui *URL*, sedangkan *SQL* melakukan pemrosesan query dengan mengakses tabel *database* relasional (Nandini, 2014).
Pada Wood et al. (2014), struktur penulisan *query SPARQL* terdiri dari:

1. *PREFIX* (*namespace prefix*)
2. *SELECT* (mendefinisikan *field* yang diinginkan)
3. *FROM* (menspesifikasikan *dataset* yang diinginkan)
4. *WHERE* {
   (Mendeskripsikan kriteria untuk menghasilkan *output* yang diinginkan. Kondisi ini dinyatakan dalam bentuk *query triple pattern*)
   }
5. *ORDER BY*, *LIMIT*, *GROUP BY*, *HAVING*, *OFFSET* dan lain-lain (*modifier* untuk menghasilkan *output* yang diinginkan)

```
PREFIX event: <http://purl.org/NET/c4dm/event.owl#>
PREFIX mo: <http://purl.org/ontology/mo/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>

SELECT ?title ?performer ?time ?place
WHERE { ?x event:place ?place
        FILTER regex(?place, "^Jakarta")
        ?x dc:title ?title.
        ?x mo:performer ?performer.
        ?x event:time ?time.
}
```

Gambar 2.5. Contoh *SPARQL Query*

2.10. *Apache Jena Fuseki*

Apache Jena Fuseki adalah sebuah *SPARQL server* yang ringan untuk menyimpan data *RDF*, memproses *query SPARQL*, melakukan *update* protokol terhadap data *RDF* menggunakan *SPARQL Graph Store HTTP Protocol*, dan mengirim hasil *query SPARQL* melalui *HTTP* (Wood et al., 2014). Dalam implementasinya, *Apache Jena Fuseki* dapat dijalankan sebagai layanan sistem operasi, aplikasi web Java (*file WAR*), atau *standalone server*.

Pemrosesan data pada *Jena* umumnya diakses dari aplikasi melalui *Java API*, dimana aplikasi mengirim data ke *Jena* melalui *internet*. *Fuseki* merupakan *data publishing server* yang berfungsi untuk menampilkan dan mengubah model *RDF* melalui *web* menggunakan *SPARQL* dan *HTTP*.

2.11. Evaluasi Kinerja Sistem

Matriks yang biasa digunakan dalam evaluasi pada sistem *Information Extraction* adalah *precision* dan *recall* (Makhoul *et al*. 2009). *Precision* merupakan jumlah item yang telah di identifikasi dengan benar atau relevan dari jumlah total item yang di identifikasi, sedangkan *recall* yaitu rasio jumlah item relevan yang ditemukan dengan total jumlah item dalam kumpulan item yang dianggap relevan. *F-score* adalah pengukuran kinerja yang merupakan kombinasi nilai *precision* dan *recall*. Perhitungan

menggunakan rumus perhitungan *precision*, *recall* dan *f-score* pada persamaan 2.1, 2.2 dan 2.3 (Baeza-Yates & Ribeiro-Neto,1999).

$$Precision = \frac{TP}{TP+FP} \tag{2.1}$$

$$Recall = \frac{TP}{TP+FN} \tag{2.2}$$

$$F\text{-}score = \frac{2 \times precision \times recall}{precision + recall} \tag{2.3}$$

Keterangan:

TP = item relevan yang ditemukan (*true positive*)

FP = item tidak relevan yang ditemukan (*false positive*)

FN = item tidak relevan yang tidak ditemukan (*false negative)*

2.12. Penelitian Terdahulu

Setiyoaji et al (2017) menggunakan *hidden markov model* dan algoritma Viterbi dalam melakukan pengenalan entitas bernama atau *Named Entity Recognition* (NER) untuk mengenali nama entitas pada teks tanaman obat diantaranya nama tanaman, tempat tanaman, kandungan zat, dan kegunaan tanaman. Pada penelitian ini dilakukan 3 pengujian dengan masing-masing mempunya skenario yang berbeda. Pada pengujian pertama dengan skenario data latih sebagai data uji menghasilkan nilai rata-rata *precision*, *recall* dan *f-measure* tertinggi yaitu masing-masing 056064, 07402 dan 05447, pengujian kedua data dengan skenario data latih tidak sebagai data uji mendapatkan rata-rata *f-measure* 0.5109 , pada pengujian ketiga dengan skenario penambahan tiga *POS tagging* yaitu *precision*, *noun*, dan *verb* mendapatkan rata-rata *f-measure* terendah yaitu 0.4908. Secara keseluruhan pengenalan entitas nilai yang paling rendah pada skenario ketiga pada entitas nama dengan *f-measure* 0.4100, dan yang paling tinggi pada skenario pertama pada entitas guna dengan *f-measure* 0.7255.

Ismaya (2014) menggunakan *rule-based classification*, dan metode preprocessing yang digunakan adalah *POS* Tagging untuk melakukan ekstraksi informasi dari sekumpulan dokumen Laporan Hasil Pemeriksaan (LHP) Badan Pemeriksa Keuangan (BPK) RI. Akurasi dari data latih dan data uji masing-masing

adalah 98,27% dan 89,77%. Penurunan akurasi antara lain disebabkan oleh penggunaan aplikasi pdf2text yang tidak memberikan hasil konversi yang identik atara data input dan output, serta penggunaan metode wordmatch untuk klasifikasi.

Permadi (2009) menggunakan *Partial Tree Alignment* (DEPTA). Dengan metode ini halaman web akan dimodelkan dengan struktur tree sebelum akhirnya di ekstraksi. Sistem DEPTA menghasilkan recall 97%, precision 37%, the harmonic mean 50% dan the e-measure 61%. Namun sistem ini hanya dapat berjalan baik pada bentuk website yang mempunyai banyak blok tabel, misalnya pada website komersial. Sebaliknya jika pada bentuk website berita, sistem ini tidak akan berjalan dengan baik.

Penelitian yang dilakukan oleh Susanti (2015) dalam ekstraksi informasi konten web menggunakan pendekatan berbasis ontologi yang disebut *Ontology-Based Information Extraction* (OBIE). Penelitian ini dilakukan dengan tujuan untuk menyediakan konten semantik untuk web semantik dan dapat melakukan proses ekstraksi secara *semi-automatic* (metode yang dapat meminimalkan keterlibatan manusia dalam proses ekstraksi). Sistem ekstraksi dikembangkan dengan menggunakan arsitektur *General Architecture for Text Engineering* (GATE). Extraction rules ditulis ke dalam format *Java Annotation Pattern Engine* (JAPE) dan diinterpretasikan dalam GATE. Sistem ini diuji coba menggunakan dataset “Lonely Planet”. Dari hasil evaluasi pengujian menggunakan *Corpus Benchmark Tools* (CBT) pada GATE menunjukkan bahwa hasil *Recall* 62%, *Precision* 73% dan *F-measure* 67%.

Riaddy (2016) menggunakan pendekatan *Supervised Learning* dalam melakukan ekstraksi informasi pada makalah ilmiah dengan menggabungkan tiga algoritma *supervised learning* yaitu *Support Vector Machine* (SVM), *Naïve Bayes*, dan *Random Forest.* Implementasi dari ketiga *classifier* tersebut pada penelitian ini menggunakan *library* dari Wakaito Environment for Knowledge Analysis (WEKA). Informasi yang diekstraksi dari makalah ilmiah pada penelitian ini yaitu judul, nama penulis, afiliasi, alamat, *email* dan *website*. Dari hasil penelitian yang dilakukan dapat disimpulkan bahwa kombinasi fitur yang tepat dapat meningkatkan akurasi *classifier supervised learning* yaitu judul dengan *word accuracy* data *training* 97.207% dan data *testing* 97.861% menggunakan *classifier Random Forest*, nama penulis dengan *word accuracy* data *training* 97.061% dan data *testing* 97.995% menggunakan SVM, afiliasi dengan *word accuracy* data *training* 96.505% dan data *testing* 96.390% menggunakan *Random Forest*, alamat dengan *word accuracy* data *training* 99.259% dan data *testing*

99.733% menggunakan SVM, *email* dengan *word accuracy* data *training* 99.947% dan data *testing* 99.866% menggunakan *Random Forest* dan *website* dengan *word accuracy* data *training* 99.987% dan data *testing* 100% menggunakan gabungan antara SVM, *Naïve Bayes*, dan *Random Forest*.

Tabel 2.4. Penelitian Terdahulu

Peneliti	Judul Penelitian	Metode	Keterangan
Setiyoaji et al, 2017	*Named Entity Recognition* Menggunakan *Hidden Markov Model* dan Algoritma Viterbi pada Teks Tanaman Obat	*Hidden Markov Model* (HMM) dan Algoritma Viterbi	Algoritma pada penelitian ini menggunakan HMM dan algoritma Viterbi
Ismaya, 2014	Algoritma Ekstraksi Informasi Berbasis Aturan	*Rule-based classification, POS Tagging*	Menggunakan metode *Rule-based classification* dan metode preprocessing menggunakan *POS Tagging*
Permadi, 2009	Ekstraksi Data pada Halaman Web Menggunakan Partial Tree Alignment (DEPTA)	*Partial Tree Alignment*	Menggunakan metode *Partial Tree Alignment* (DEPTA)
Susanti, 2015	Ekstraksi Informasi Konten Web Menggunakan Pendekatan berbasis Ontologi	*Ontology Based Information Extraction* (OBIE)	Menggunakan ontologi yang disebut *Ontology-Based Information Extraction* (OBIE)
Riaddy, 2016	*Information Extraction on Scientific Papers with Supervised Learning Approach*	*Supervised Learning*	Menggunakan metode Supervised Learning dengan menggunakan tiga algoritma yaitu SVM, *Naïve Bayes*, dan *Random Forest*.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

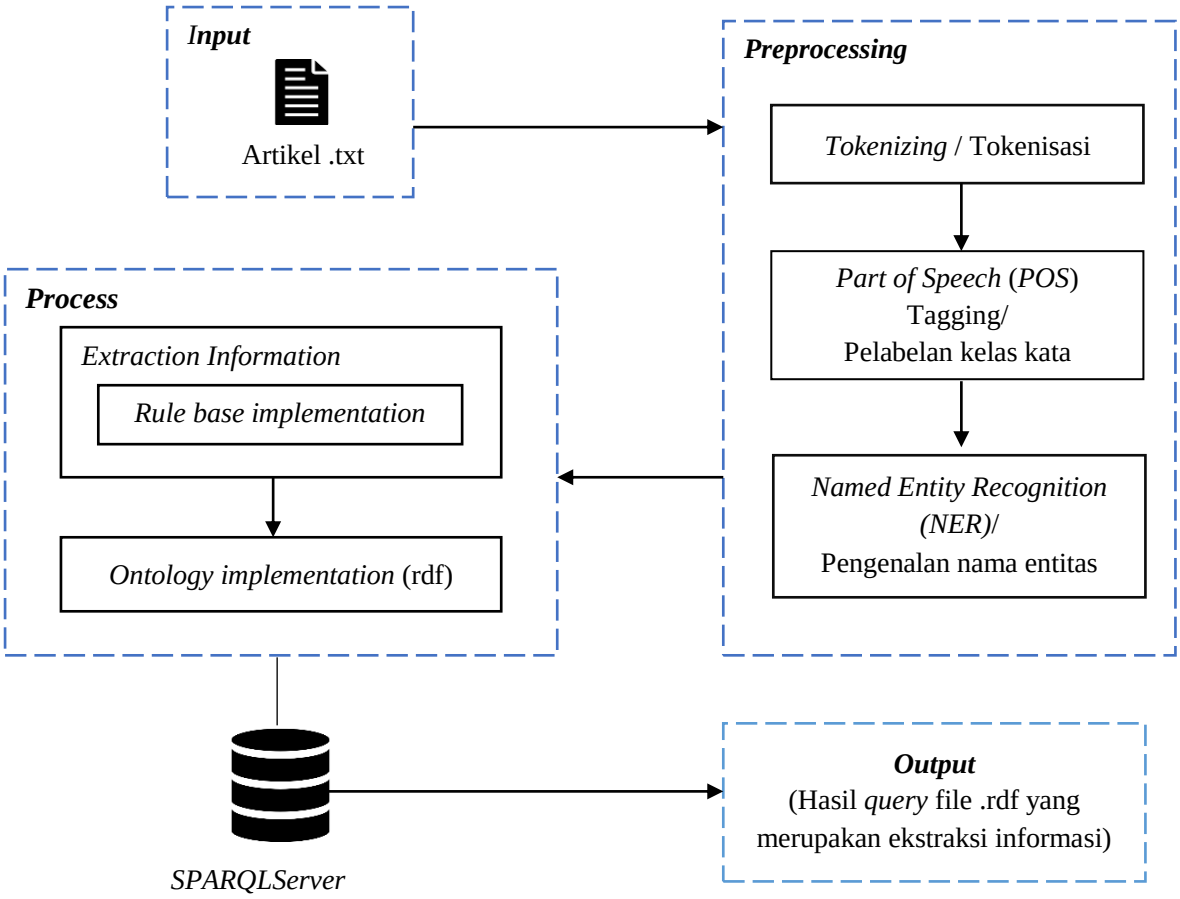
Bab ini akan menjelaskan tentang implementasi metode yang digunakan dalam sistem ekstraksi informasi artikel musik. Adapun tahapan yang dibahas pada bab ini yaitu tahap analisis dan tahap perancangan sistem. Analisis terhadap data yang digunakan dan analisis terhadap metode yang digunakan pada langkah-langkah pemrosesan data akan dibahas pada tahap analisis. Perancangan tampilan antarmuka sistem akan dibahas pada tahap perancangan sistem.

3.1 Data yang Digunakan

Pada penelitian ini digunakan 2 macam data yaitu data pelatihan dan data uji. Data pelatihan yang digunakan yaitu berupa korpus. Korpus yang digunakan pada proses *part of speech tagging* berisi teks bahasa indonesia yang sudah memiliki kelas katanya, dimana korpus tersebut merupakan hasil penelitian *POS tagging* sebelumnya yang juga menggunakan metode *hidden markov model* (Wicaksono & Purnawarianti, 2010). Sedangkan pada proses *named entity recognition* digunakan korpus dari penelitian *named entity recognition*yang juga menggunakan metode *hidden markov model* (Fachri, 2009). Data uji yang digunakan pada penelitian ini diambil secara manual tidak menggunakan proses *crawling* dikarenakan penelitian berfokus kepada proses ekstraksi informasi artikel. Data uji yang digunakan pada pelatihan ini yaitu 50 artikel yang diambil dari media online seperti sindonews.com, bintang.com, metrotvnews.com, tribunnews.com.

3.2 Analisis Sistem

Pada tahap ini dilakukan perancangan sistem untuk menyelesaikan permasalahan yang terdapat di dalam tahap analisis.Arsitektur umum dari perancangan yang digunakan pada penelitian ini ditunjukkan pada Gambar 3.1.



Gambar 3.1. Arsitektur Umum

Metode yang diajukan penulis untuk melakukan ekstraksi informasi artikel musik terdiri dari beberapa proses. Proses-proses yang dilakukan yaitu *pre-processing*, pada proses ini terlebih dahulu dilakukan pemecahan kalimat menjadi kata-kata atau disebut juga tokenisasi, proses ini dilakukan karena selanjutnya teks akan melewati proses *part of speech tagging* untuk mengetahui kelas kata yang dimiliki masing-masing kata. Setelah itu dengan menggunakan probabilitas pada proses *named entity recognition* maka sistem akan mengetahui apakah suatu kata termasuk ke dalam entitas tempat, entitas orang, entitas lokasi, entitas organisasi dan entitas lainnya.

Setelah dilakukan proses *part of speech tagging* dan *named entity recognition* maka dilakukan proses ekstraksi informasi untuk mendapatkan hasil ekstraksi yaitu nama acara, nama artis, tempat dan tanggal. Pada proses ekstraksi informasi dilakukan implementasi *rule* yang telah didefinisikan berdasarkan hasil *named entity recognition* untuk mendapatkan hasil dari tempat dan tanggal, sedangkan hasil yang berupa nama acara dan nama artis didapat berdasarkan proses *named entity recognition*. Hasil akhir

dari ekstraksi tersebut dibuat berdasarkan ontologi musik dan proses akhir dari sistem yaitu hasil ekstraksi dapat ditampilkan dan disimpan dalam format .rdf.

Adapun tahapan keseluruhan proses adalah sebagai berikut:

3.2.1. Input

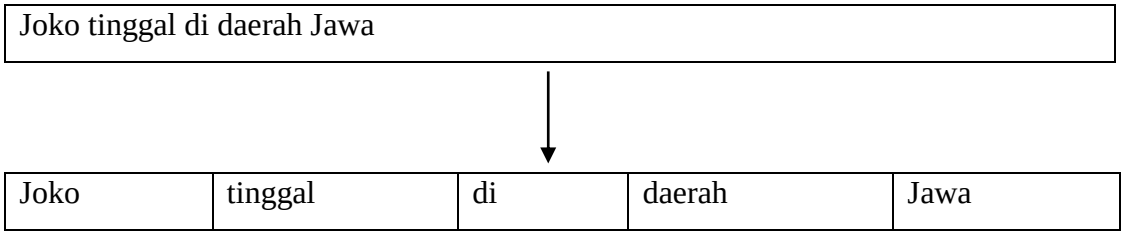
Tahapan awal yang dilakukan adalah menginput file berupa teks artikel musik khususnya yang berhubungan dengan acara musik. File yang dimasukkan adalah yang berekstensi .txt. Teks yang dimasukkan harus sesuai dengan aturan tanda baca yang ditentukan dan tidak terdapat kesalahan penulisan agar sistem dapat berjalan dengan optimal dan ekstraksi informasi didapat lebih akurat.

3.2.2. Pre-processing

Tahap ini bertujuan untuk memproses artikel mentah menjadi artikel yang siap untuk diproses lebih lanjut. Adapun beberapa tahapan-tahapannya yaitu:

3.2.2.1. Tokenizing

Pada proses *tokenizing* dilakukan pemecahan input teks pada paragraf ataupun kalimat yang terdapat di dalam file yang diinput. Biasanya yang menjadi acuan pemisah antar kata adalah karakter spasi atau tanda baca. Meskipun kata-kata dalam kalimat ataupun paragraf tersebut dipisah namun urutan katanya masih dipertahankan.



Gambar 3.2. Contoh Hasil *Tokenizing*

3.2.2.2. POS Tagging

Setelah dilakukan *tokenizing*, maka sistem akan melakukan proses *POS tagging*. Pada proses *POS tagging* akan dilakukan pelabelan kelas kata pada sekumpulan teks yang telah diinput ke dalam sistem. Adapun jenis label kata tersebut yaitu kata kerja (*verb*), kata benda (*noun*), kata sifat (*adjective*), kata keterangan (*adverb*), kata depan

(*preposition*) dan lain sebagainya. Pada proses *POS tagging* ini akan digunakan modul *POS tagging* untuk bahasa Indonesia dimana proses akan dilakukan dengan metode *hidden markov model* (HMM) dalam melakukan prediksi dan penentuan label kata (Wicaksono & Purnawarianti, 2010).

Joko/NNP tinggal/VBI di/IN daerah/NN Jawa/NN

Gambar 3.3 Contoh Hasil *POS Tagging*

3.2.2.3. Named Entity Recognition (NER)

Proses *named entity recognition* atau pengenalan entitas bernama merupakan proses yang berguna untuk mendeteksi entitas yang dimiliki oleh kata pada suatu teks. Namun, proses pengenalan entitas sangat bergantung pada tingkat kebenaran *case* pada teks yang diiput, jika tingkat kebenaran *case* yang di input rendah, maka proses NER akan kesulitan untuk mengenali entitas pada teks tersebut. NER menentukan entitas dari setiap kata menggunakan probabilitas yang didapatkan dari mengobservasi kelas kata pada kata tersebut, kelas kata sebelum, dan kelas kata sesudahnya. Dilakukannya pengenalan entitas ini sangat membantu sistem ekstraksi informasi untuk mendapatkan hasil yang baik dan tepat. Entitas yang didapatkan dari tahapan *named entity recognition* ini terdiri dari nama orang, nama tempat, waktu, organisasi dan lainnya. Untuk tahapan *named entity recognition* ini, akan digunakan metode *hidden markov model* (Fachri, 2014).

<PERSON>Joko<PERSON>tinggal di daerah <LOCATION>Jawa<LOCATION>

Gambar 3.4 Contoh Hasil *Named Entity Recognition*

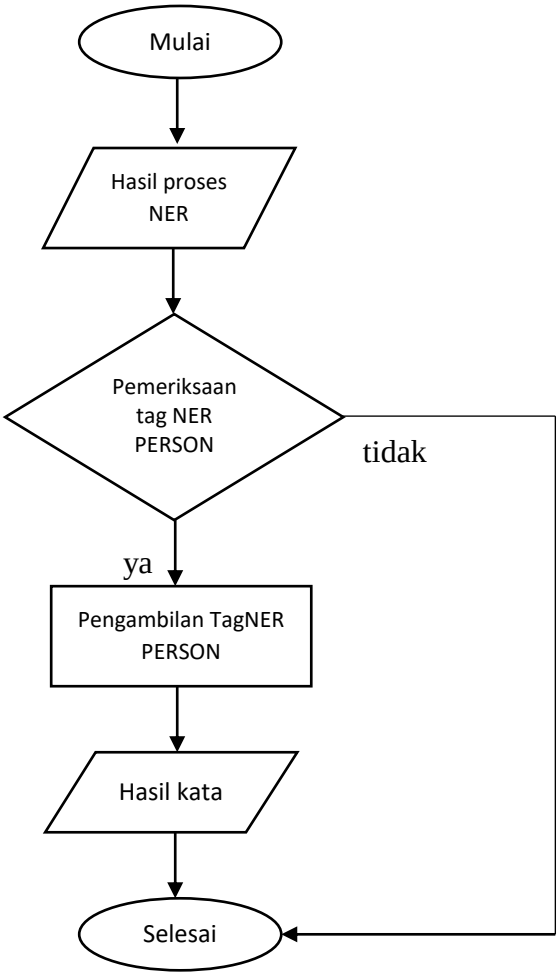
3.2.3. Process

3.2.3.1. Extraction Information

Pada tahap ini dilakukan ekstraksi informasi dari teks yang mengambil data berupa nama, nama acara, waktu, dan tempat. Dalam tahap ini juga dilakukan pengimplementasian metode *rule based* untuk mendapatkan data waktu dan tempat.

1. Pengambilan Nama

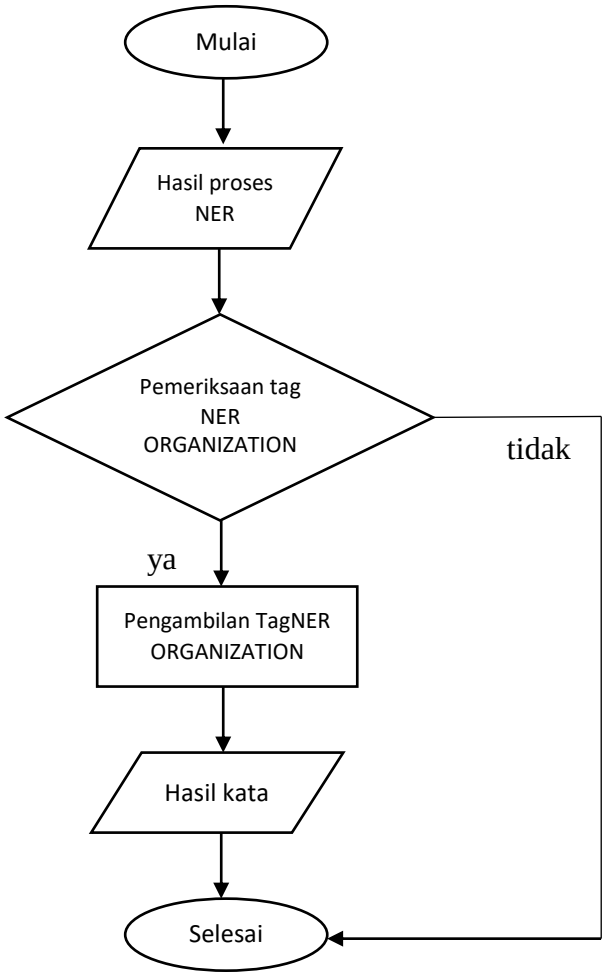
Aturan yang dibuat untuk pengambilan nama didapatkan dari proses *named entitiy recognition*, melalui proses NER sistem telah menentukan entitas tag PERSON dalam artikel, melalui tag PERSON penulis akan menentukan bahwa PERSON adalah nama. Dilihat bahwa PERSON utama dari sebuah artikel terdapat pada awal paragraf maka sistem akan mengambil tag tersebut, pada sistem ini penulis membatasi maksimum panjang string untuk nama yaitu 4 kata. Contohnya seperti "Festival SHVR Ground akan mengundang penyanyi Khalid. Ini adalah kali pertama Khalid datang ke Indonesia" , maka sesuai contoh diatas, melalui proses NER akan didapat tag PERSON adalah Khalid, kemudian sistem akan mengambil bahwasanya Khalid adalah nama.



Gambar 3.5. Perancangan Proses Pengambilan Entitas Nama

2. Pengambilan Title

Pada proses pengambilan title acara dari artikel musik, dilihat bahwa acara dalam sebuah artikel dikategorikan sebagai tag ORGANIZATION dalam proses *named entity recognition.* Maka seperti aturan pengambilan nama sebelumnya, pengambilan *title* acara dalam artikel musik, menggunakan hasil yang didapat dari proses NER, contohnya "Festival SHVR Ground akan mengundang penyanyi Khalid" , dari kalimat tersebut pada proses *named entity recognition* akan mendapatkan bahwasanya Festival dan SHVR Ground merupakan tag ORGANIZATION, kemudian penulis akan menggabungkan hasil dari kedua tag tersebut menjadi satu.



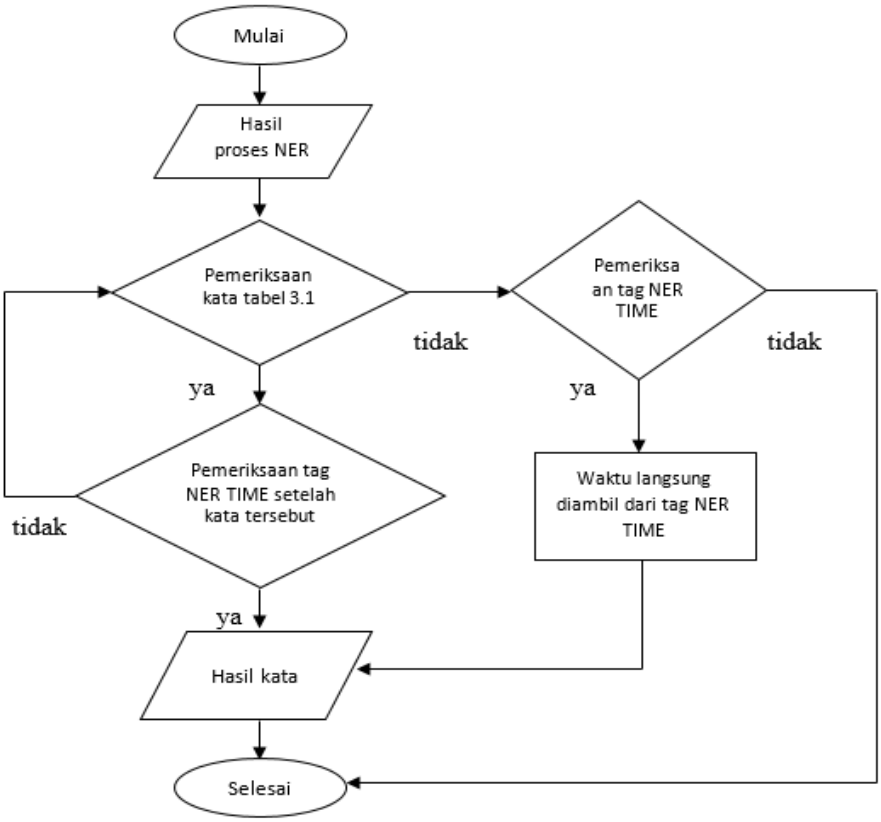
Gambar 3.6. Perancangan Proses Pengambilan Entitas Title / Nama Acara

3. Pengambilan Waktu (Time)

Untuk aturan pengambilan waktu, dalam proses *named entity recognition* terdapat tag TIME, maka dari tag tersebut penulis akan mendapatkan hasil waktu. Namun untuk pengambilan tag waktu penulis menambahkan aturan atau *rule* untuk mendapatkan waktu yang lebih tepat. Aturan yang dibuat yaitu pemeriksaan apabila ada kata-kata tertentu sebelum tag TIME yaitu :

Tabel 3.1. Kata yang diperiksa sebelum tag TIME

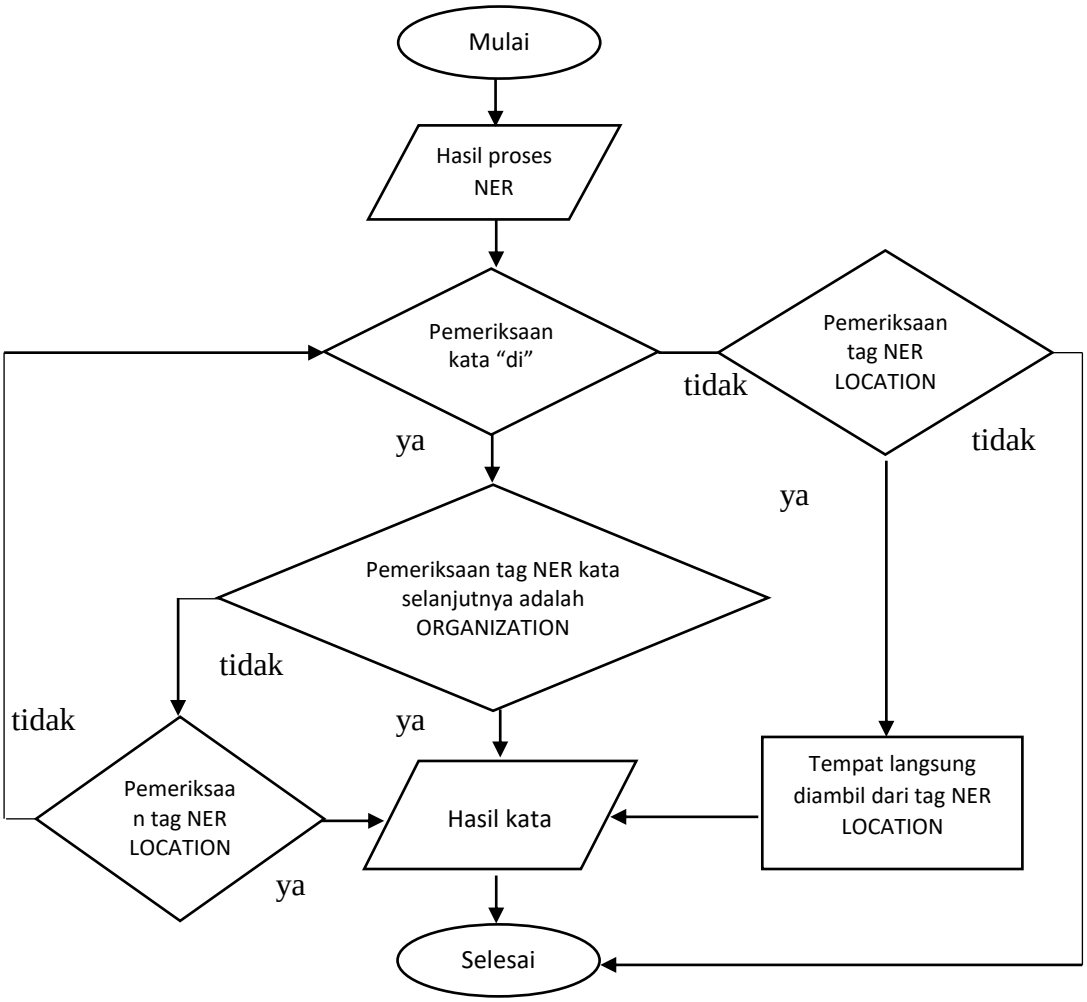
No	Kata
1	Diselenggarakan
2	Dilaksanakan
3	Digelar
4	Berlangsung
5	Pada



Gambar 3.7. Perancangan Proses Pengambilan Entitas Waktu

4. Aturan Pengambilan Tempat (Place)

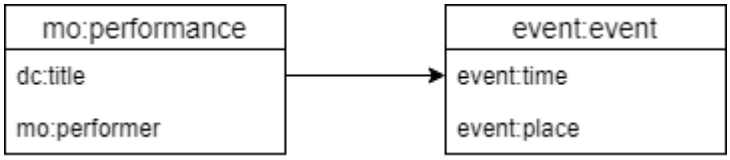
Setelah melakukan proses *named entity recognition* sistem telah mendapatkan nama dari tempat dalam artikel dengan tag LOCATION. Namun penulis menambahkan aturan atau *rule* untuk mendapatkan hasil tempat yang benar. Adapun untuk pengambilan entity tempat dilakukan pengecekan apakah ada kata “di” yang terdapat di depan tag ORGANIZATION pada hasil NER, pengecekan menggunakan tag ORGANIZATION karena nama tempat atau *venue* untuk melakukan sebuah acara biasa di tag dengan tag ORGANIZATION, tetapi apabila tidak terdapat tag tersebut, maka sistem akan memeriksa apakah di depan kata “di” merupakan tag LOCATION, kalau hasilnya benar maka akan dimasukkan ke dalam hasil tempat.



Gambar 3.8.Perancangan Proses Pengambilan Entitas Tempat

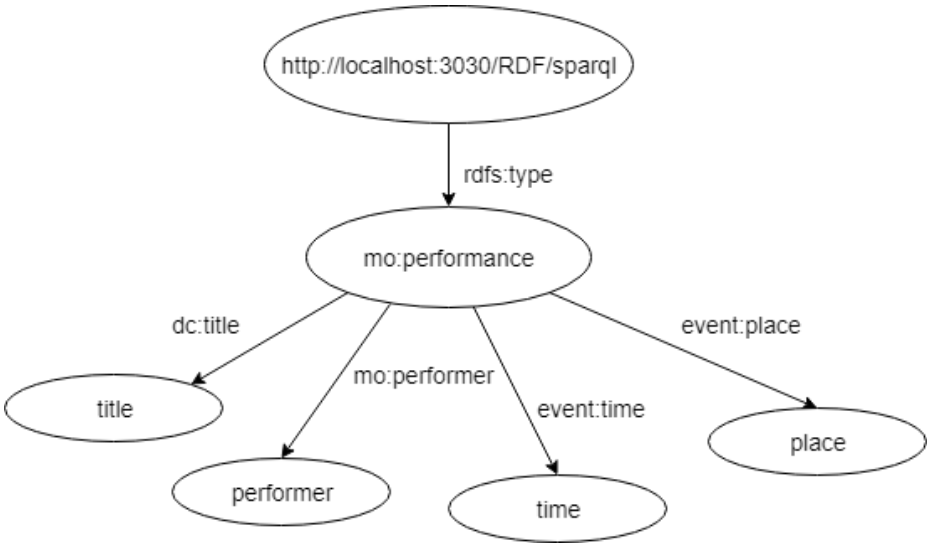
3.2.3.2. Penerapan Ontologi

Pada ekstraksi informasi dilakukan penerapan ontologi musik terhadap hasil ekstraksi informasi artikel. Hasil-hasil tersebut akan disimpan dalam bentuk RDF/XML yang disesuaikan dari ontologi musik. Ontologi musik yang dipakai sesuai dengan data yang diambil yaitu nama artis, tanggal acara, tempat acara dan nama acara.Konsep ontologi musik yang ditunjukkan dalam graph dan relasinya yang diambil untuk penelitian ini ditunjukkan pada Gambar 3.9.



Gambar 3.9. Graph dan Relasi Konsep Ontologi Musik pada Penelitian ini

Seperti ditunjukkan pada tabel 3.9. ontologi musik sebagian besar dipengaruhi oleh beberapa ontologi salah satu yang dipakai pada penelitian ini adalah ontologi *event*, pada graph tersebut ditunjukkan relasi dari ontologi musik (*mo:performance*) dan ontologi *event* (*event:event*) yaitu *mo:performance* merupakan *sub-class* dari *event:event*. Sedangkan contoh RDF graph pada penelitian ini yang merepresentasikan hasil dari ekstraksi informasi pada artikel musik yang berhubungan dengan konser ditunjukkan pada Gambar 3.10



Gambar 3.10. Contoh RDF Graph Dari Hasil Ektraksi Informasi

Contoh hasil ekstraksi informasi yang berupa file .rdf dalam bentuk RDF/XML yang disesuaikan dengan ontologi musik ditunjukkan pada Gambar 3.11.

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:vs="http://www.w3.org/2003/06/sw-vocab-status/ns#"
    xmlns:timeline="http://purl.org/NET/c4dm/timeline.owl#"
    xmlns:time="http://www.w3.org/2006/time#"
    xmlns:event="http://purl.org/NET/c4dm/event.owl#"
    xmlns:foaf="http://xmlns.com/foaf/0.1/"
    xmlns:frbr="http://purl.org/vocab/frbr/core#"
    xmlns:wot="http://xmlns.com/wot/0.1/"
    xmlns:bio="http://purl.org/vocab/bio/0.1/"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:mo="http://purl.org/ontology/mo/">

<mo:Performance>
<dc:title>title</dc:title>
<mo:performer>performer</mo:performer>
<event:time>time</event:time>
<event:place>place</event:place>
</mo:Performance>

</rdf:RDF>
```

Gambar 3.11. Contoh Hasil Ekstraksi Informasi Dalam Bentuk RDF Berdasarkan Ontologi Musik

3.2.4. Database Server

Penggunaan database server diimplementasikan pada sistem hanya untuk menampilkan hasil akhir dari proses ekstraksi informasi. Hasil dari ekstraksi informasi akan ditampilkan dengan menggunakan *database server* dapat diilustrasikan pada Gambar 3.2. dimana *database* terdiri dari file-file .rdf yang merupakan hasil dari ekstraksi informasi yang telah disimpan sebelumnya.

Tabel 3.2. Model D*atabase* File RDF Dalam Bentuk N-Triples

Database	**Model Data (*N-Triples*)**
File	*{ID, #performer, performer}*
	{ID, #time, time}
	{ID, #place, place}
	{ID, #title, title}

File-file rdf hasil dari ekstraksi informasi tersimpan dalam bentuk *N-Triples* yang terdiri dari *{ID, #performer, performer}*, *{ID, #time, time}*, *{ID, #place, place}* dan *{ID, #title, title}*. Semua file rdf yang di upload akan dimasukkan dalam satu *database* atau dataset pada *SPARQL server*. *SPARQL server* merupakan *server* yang menyimpan data untuk dapat dipanggil ke dalam sistem aplikasi. Agar data dapat ditampilkan ke dalam sistem maka pemanggilan data dilakukan dengan menggunakan *SPARQL query*. Ketika pemanggilan data dilakukan menggunakan *SPARQL query*, *SPARQL server* akan melakukan pencarian ke dalam *database* dan sistem akan menampilkan data ke dalam sistem.

3.2.5. Output

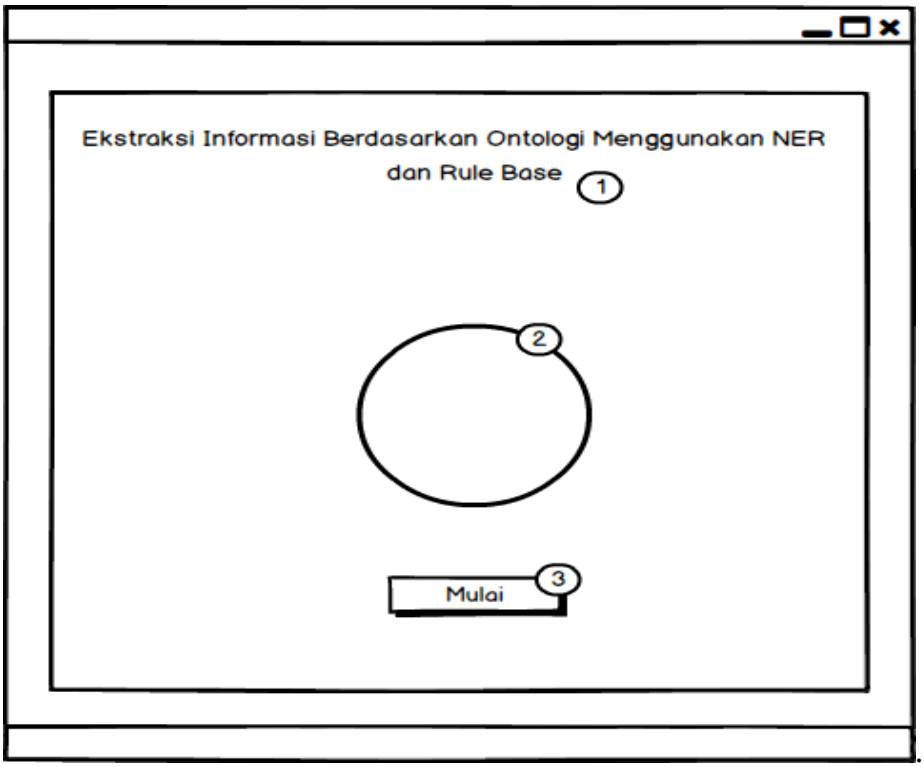
Pada tahapan ini dilakukan penampilan hasil dari ekstraksi informasi teks yang diproses pada sistem ini. Hasil yang ditampilkan merupakan file .rdf yang sudah di masukkan ke dalam *SPARQL server* kemudian dengan menggunakan *SPARQL query* dipanggil ke dalam sistem untuk ditampilkan.

3.3 Perancangan Sistem

Dalam membangun sistem pada penelitian ini, diperlukan adanya perancangan antarmuka (*interface*), yaitu gambaran umum perancangan setiap tampilan yang terdapat pada sistem yang dibangun.Perancangan sistem bagian depan (*front-end*) pada penelitian ini menggunakan *Graphical User Interface* (GUI). Sistem bagian depan dirancang untuk menjalankan proses pada penelitian ini dan menampilkan hasil dari proses.

3.3.1. Rancangan Tampilan Halaman Utama

Pada rancangan tampilan halaman utama ini berfungsi untuk menampilkan halaman utama yang menampilkan button untuk menuju ke halaman proses



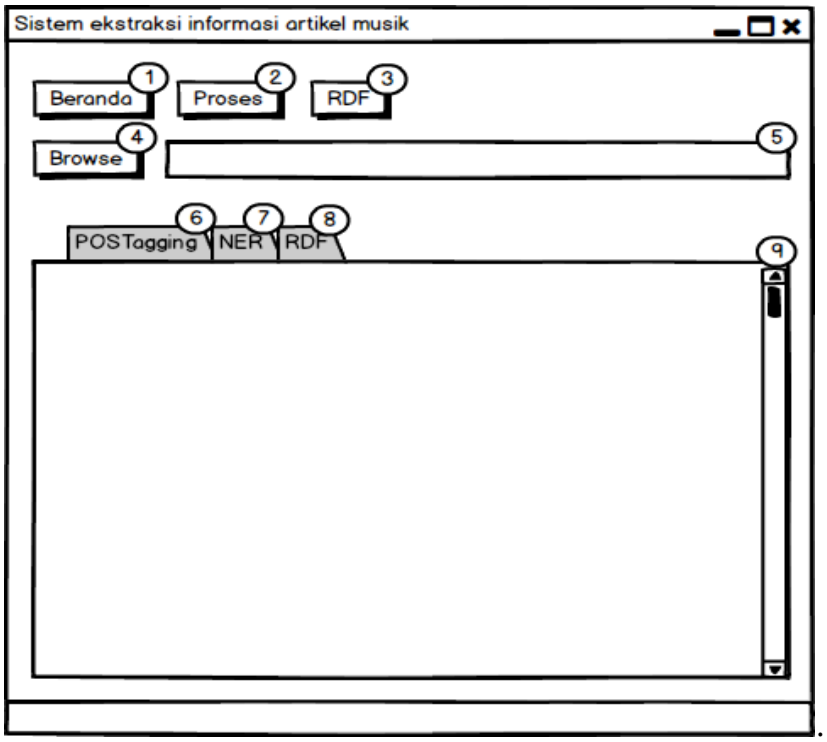
Gambar 3.12. Rancangan Tampilan Halaman Utama

Keterangan:

1) Judul, merupakan nama sistem
2) Logo, merupakan logo USU
3) Tombol Mulai adalah tombol perintah untuk masuk kehalaman Proses

3.3.2. Rancangan Tampilan Halaman Proses

Pada rancangan tampilan halaman proses ini berfungsi untuk menampilkan halaman proses dalam melakukan ekstraksi informasi dan mengetahui hasil proses *POStagging* dan NER serta ekstraksi dalam bentuk RDF



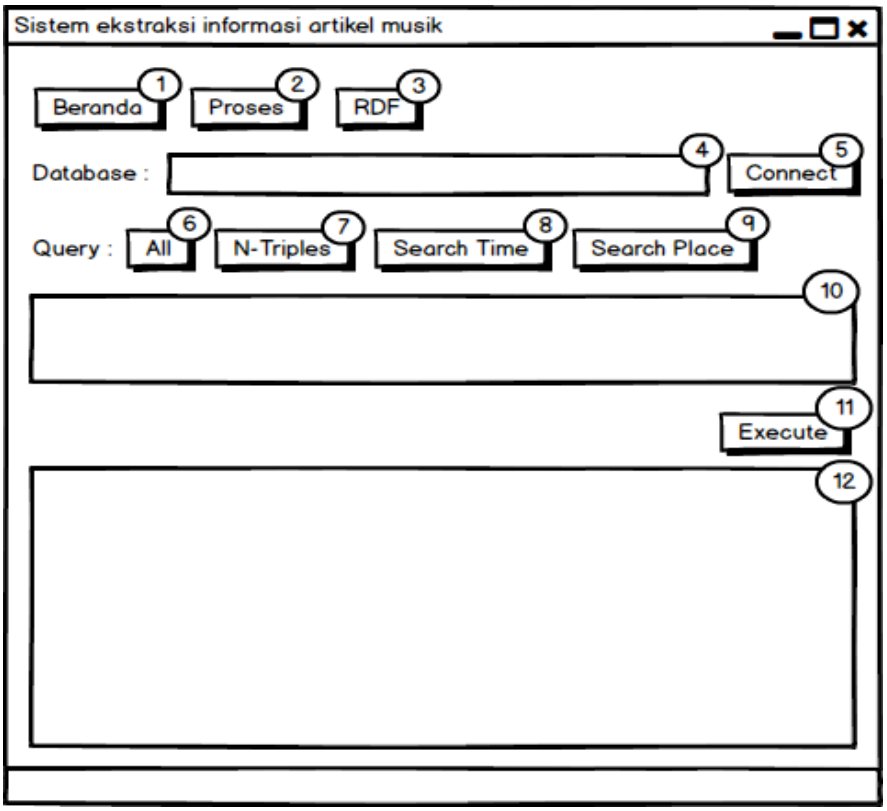
Gambar 3.13. Rancangan Tampilan Halaman Proses

Keterangan:

1) Tombol perintah untuk menampilkan halaman utama
2) Tombol perintah untuk menuju ke halaman proses program
3) Tombol perintah untuk menuju ke halaman RDF
4) Tombol perintah untuk memilih file artikel yang akan diperiksa
5) *Text box*untuk menampilkan nama file yang akan diperiksa
6) Tombol perintah untuk menunjukkan ke tampilan hasil *POS tagging*
7) Tombol perintah untuk menunjukkan ke tampilan hasil NER
8) Tombol perintah untuk menunjukkan ke tampilan hasil ekstraksi dalam bentuk RDF
9) *Text box* untuk menampilkan hasil dari proses *POS tagging*, NER dan juga ekstraksi informasi dalam bentuk RDF

3.3.3. Rancangan Tampilan Halaman RDF

Pada rancangan tampilan halaman RDF ini berfungsi untuk menampilkan output dari hasil ekstraksi yang awalnya berupa file RDF untuk ditampilkan dalam sistem



Gambar 3.14. Rancangan Tampilan Halaman RDF

Keterangan:

1) Tombol perintah untuk menampilkan halaman utama
2) Tombol perintah untuk menuju ke halaman proses program
3) Tombol perintah untuk menuju ke halaman RDF
4) *Text box* untuk menampilkan nama *localhost database* yang dipakai
5) Tombol perintah untuk melakukan koneksi ke *database*
6) Tombol perintah untuk menampilkan contoh *SPARQL query* yang mempunyai fungsiuntuk menunjukkan semua hasil ekstraksi informasi sesuai artikel masing-masing.
7) Tombol perintah untuk menampilkan contoh *SPARQL query* yang mempunyai fungsiuntuk menunjukkan hasil ekstraksi informasi dengan bentuk *n-triples*

8) Tombol perintah untuk menampilkan contoh *SPARQL query* yang mempunyai fungsiuntuk menunjukkan hasil pencarian entitas *time* tertentu pada hasil ekstraksi informasi.
9) Tombol perintah untuk menampilkan contoh *SPARQL query* yang mempunyai fungsiuntuk menunjukkan hasil pencarian entitas *place* tertentu pada hasil ekstraksi informasi.
10) *Text box*, merupakan kolom isian untuk menginput *SPARQL query*
11) Tombol perintah untuk mengeksekusi perintah *SPARQL query*
12) *Text box* untuk menampilkan hasil dari eksekusi dari *SPARQL query*

BAB IV

IMPLEMENTASI DAN PENGUJIAN

Bab ini membahas hasil yang didapatkan dari implementasi dan pengujian sistem dalam melakukan ekstraksi informasi artikel musik sesuai dengan analisis dan perancangan yang telah dibahas pada Bab 3.

4.1. Implementasi Sistem

4.1.1. Spesifikasi Perangkat Keras

Adapun spesifikasi perangkat keras dan perangkat lunak yang digunakan untuk membangun aplikasi ini adalah sebagai berikut :

1. Processor Intel(R) Core(TM) i3-3110M CPU @ 2.40GHz 2.40GHz
2. Kapasitas *hard disk* 500 GB
3. RAM 4GB

4.1.2. Spesifikasi Perangkat Lunak

Selain perangkat keras, sistem juga dibuat dalam lingkungan spesifikasi perangkat lunak sebagai berikut:

1. Windows 10 Pro
2. Eclipse Oxygen (4.7.3a)
3. Java version 8.0.1810.13
4. Google Chrome v68.0.3440.106
5. *Apache Jena Fuseki* 3.8.0

4.1.3. Implementasi Perancangan Antarmuka

Adapun implementasi dari perancangan antarmuka yang telah dibangun pada sistem ini adalah sebagai berikut:

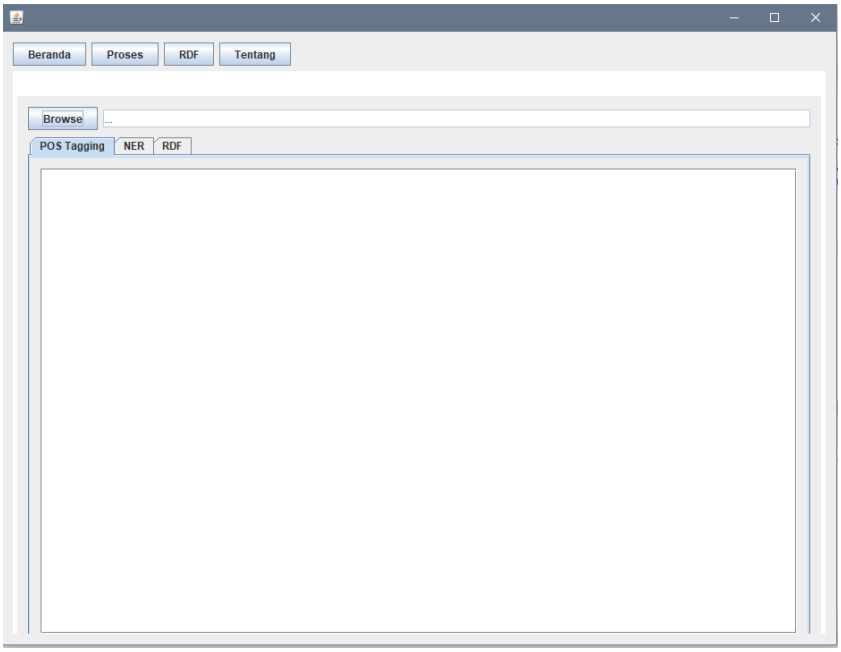
1. Halaman Utama

Halaman utama merupakan halaman yang ditampilkan pertama sekali kepada pengguna saat sistem dijalankan. Pada tampilan utama terdapat tombol Mulai untuk masuk ke dalam halaman Proses.

Gambar 4.1. Tampilan Halaman Utama

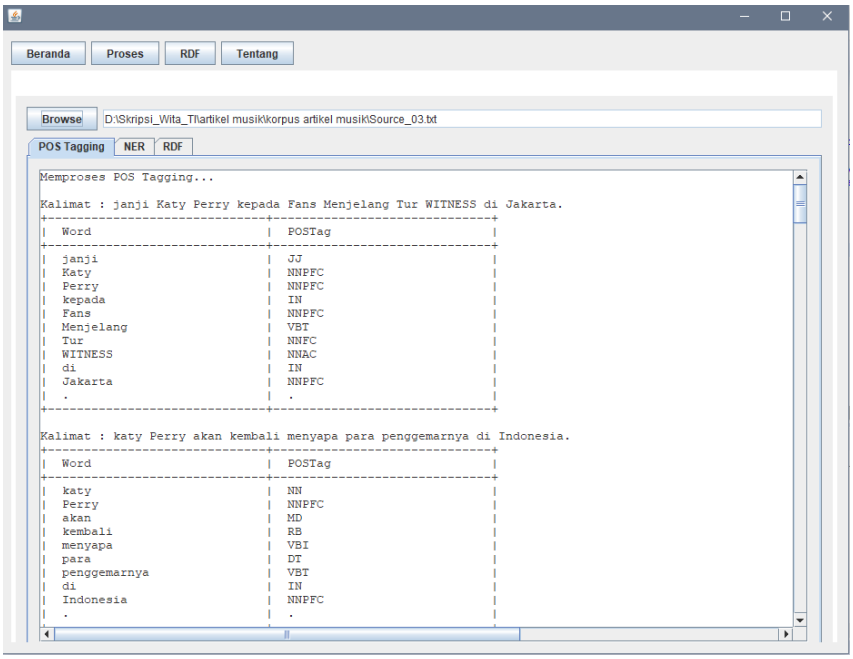
2. Halaman Proses

Adalah halaman dimana proses utama dilakukan, pada halaman ini dilakukan penginputan artikel untuk dilakukan proses *POS tagging*, *named entity recognition* dan ekstraksi informasi.

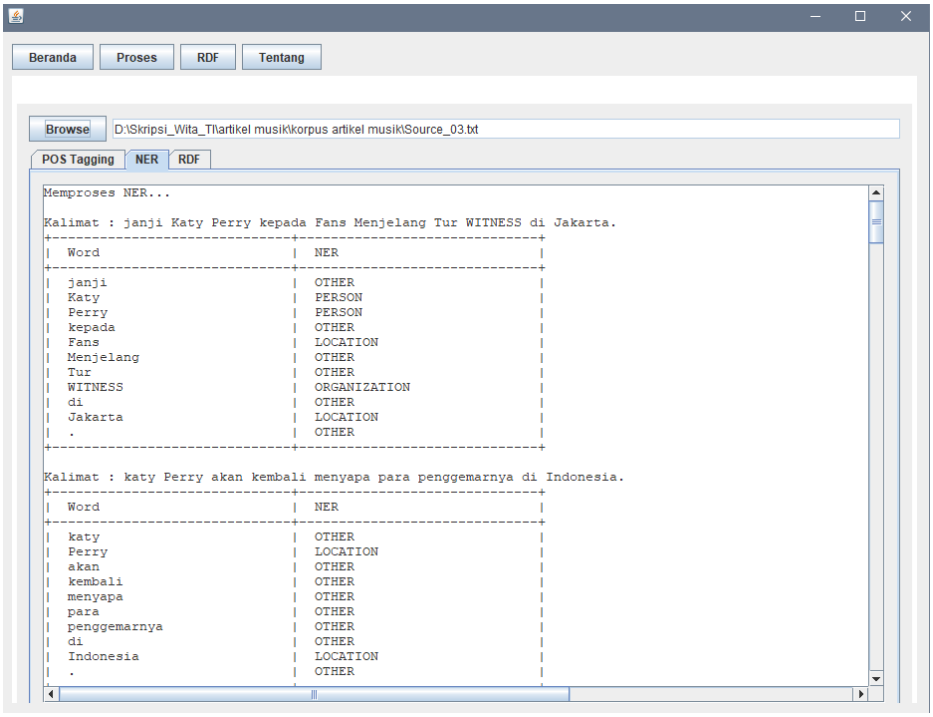


Gambar 4.2. Tampilan Halaman Proses

Setelah *user* memilih dokumen artikel yang akan dilakukan ekstraksi maka, sistem akan memulai proses ektraksi dengan memeriksa keseluruhan isi dokumen untuk didapatkan hasil ekstrasinya berdasarkan label kelas kata, label entitas dan impelementasi *rule* yang ada. Untuk hasil *POS tagging*, kata dan hasil dari *POS tagging* ditampilkan per-kalimat sesuai yang ada di dalam artikel dalam bentuk tabel. Tampilan hasil dari proses *POS tagging* atau pelabelan kelas kata dapat dilihat pada Gambar 4.3.



Gambar 4.3. Tampilan Hasil *POS Tagging*



Gambar 4.4 Tampilan Hasil *Named Entity Recognition*

Sama seperti *POS tagging*, kata dan hasil dari NER yang berupa entitas kata ditampilkan per-kalimat dalam bentuk tabel sebanyak kalimat yang terdapat di dalam artikel. Tampilan dari hasil pelabelan entitas kata atau *named entity recognition* dapat dilihat pada Gambar 4.4. proses pelabelan entitas kata berperan penting dalam proses ekstraksi informasi untuk menentukan *rule* dan juga untuk mendapatkan hasil dari ekstraksi informasi tersebut.

Setelah dilakukan proses *pre-processing* sampai proses NER, maka dilakukan proses ekstraksi informasi dan juga implementasi *rule* untuk mendapatkan hasil waktu dan tempat. Contoh diambil dari hasil NER dibawah ini:

<ORGANIZATION>Festival SHVR Ground<ORGANIZATION> akan mengundang penyanyi <PERSON> Khalid <PERSON>. <ORGANIZATION>Festival SHVR Ground<ORGANIZATION> ini akan diselenggarakan pada <TIME>4<TIME> <TIME>Mei<TIME> <TIME>2018<TIME> di <LOCATION> Jakarta <LOCATION>.

Ekstraksi informasi diawali dari pengambilan entitas *performer* atau nama artis. Untuk mendapatkan entitas *performer* menggunakan hasil pelabelan entitas nama dengan tag

PERSON, sesuai contoh NER diatas hasil yang mempunyai hasil tag PERSON adalah ‘Khalid’.

Festival SHVR Ground akan mengundang penyanyi <PERSON> Khalid <PERSON>.

Maka sistem akan menyimpannya ‘Khalid’ ke dalam memori sistem sebagai entitas *performer*.

Entitas	Hasil
Performer	Khalid

Setelah didapatkan hasil dari entitas performer maka selanjutnya adalah pengambilan entitas *title* atau nama acara. Untuk mendapatkan entitas nama acara diambil dari hasil NER yaitu tag ORGANIZATION. Sesuai contoh NER sebelumnya didapat tag ORGANIZATION yaitu:

<ORGANIZATION>Festival SHVR Ground<ORGANIZATION> akan mengundang penyanyi Khalid <ORGANIZATION>Festival SHVR Ground<ORGANIZATION> ini akan diselenggarakan pada 4 Mei 2018 di Jakarta.

Maka sistem akan menyimpan ‘Festival SHVR Ground’ ke dalam memori sistem sebagai entitas *title*. Meskipun terdapat dua hasil tag ORGANIZATION apabila hasil dari tag tersebut adalah sama maka hasil tidak akan ditampilkan secara berulang.

Entitas	Hasil
Performer	Khalid
Title	Festival SHVR Ground

Selanjutnya adalah pengambilan entitas *time*. Entitas *time* didapatkan dari hasil NER dengan tag TIME dan juga implementasi *rule based*, contoh hasil tag TIME dari NER yaitu:

Festival SHVR Ground ini akan diselenggarakan pada <TIME>4<TIME> <TIME>Mei<TIME> <TIME>2018<TIME> di Jakarta.

Pada hasil diatas terdapat tiga hasil tag TIME yang didapat dari proses NER. Untuk mendapatkan tanggal yang sesuai maka pada proses ini ditambahkan aturan yaitu dilakukan pengecekan kata sebelum tag TIME, kata-kata tersebut adalah ‘diselenggarakan’, ‘dilaksanakan’, ‘digelar’, ‘berlangsung’ dan ‘pada’. Pengecekan terhadap kata-kata ini dilakukan karena tidak semua tag TIME merupakan tag yang tepat untuk dimasukkan ke dalam entitas *time* yang menunjukkan waktu diadakannya sebuah acara musik. Dilihat dari contoh kalimat diatas terdapat kata ‘diselenggarakan’ dan ‘pada’ maka sistem akan menyimpan hasil tag TIME tersebut ke dalam sistem sebagai entitas *time*.

Entitas	**Hasil**
Performer	Khalid
Title	Festival SHVR Ground
Time	4 Mei 2018

Yang terakhir adalah pengambilan entitas *place*. Entitas *place* didapatkan dari hasil tag NER yaitu ORGANIZATION dan LOCATION dan juga implementasi *rule based*. Contoh hasil tag NER sebelumnya adalah:

Festival SHVR Ground ini akan diselenggarakan pada 4 Mei 2018 di <LOCATION>Jakarta<LOCATION>.

Pada hasil diatas terdapat tag LOCATION yaitu ‘Jakarta’, sesuai dari *rule* yang telah ditentukan adalah apabila terdapat kata ‘di’ didepan tag LOCATION kata yang mempunyai tag LOCATION tersebut akan dikategorikan ke dalam entitas *place* maka ‘Jakarta’ akan disimpan ke dalam sistem sebagai entitas *place*.

Entitas	Hasil
Performer	Khalid
Title	Festival SHVR Ground
Time	4 Mei 2018
Place	Jakarta

Hasil entitas *place* bisa juga didapatkan dari tag ORGANIZATION yaitu apabila terdapat kata ‘di’ didepan tag ORGANIZATION maka kata yang mempunyai tag ORGANIZATION tersebut akan dikategorikan ke dalam entitas *place*. Tag ORGANIZATION dipakai dalam *rule* untuk mendapatkan entitas LOCATION karena nama *venue* atau tempat untuk melaksanakan suatu acara musik biasa dikategorikan ke dalam tag ORGANIZATION dalam proses NER. Untuk memastikan bahwa tag ORGANIZATION tersebut adalah nama tempat maka ditambahkan pengecekan kata ‘di’ didepan kata ber-tag ORGANIZATION tersebut. Jadi apabila dalam suatu artikel secara bersamaan terdapat kata ‘di’ didepan tag LOCATION dan ORGANIZATION seperti contoh:

Festival SHVR Ground ini akan diselenggarakan pada 4 Mei 2018 di <ORGANIZATION>Allianz<ORGANIZATION> <ORGANIZATION>Ecopark<ORGANIZATION> <LOCATION>Ancol<LOCATION>, <LOCATION>Jakarta<LOCATION>.

Maka hasil yang disimpan dalam sistem yaitu:

Entitas	Hasil
Performer	Khalid
Title	Festival SHVR Ground
Time	4 Mei 2018
Place	Allianz Ecopark Ancol Jakarta

Setelah didapat hasil ekstraksi empat entitas *performer*, *title*, *time* dan *place* maka hasil-hasil tersebut dimasukkan ke dalam template RDF yang diambil berdasarkan ontologi musik contohnya adalah:

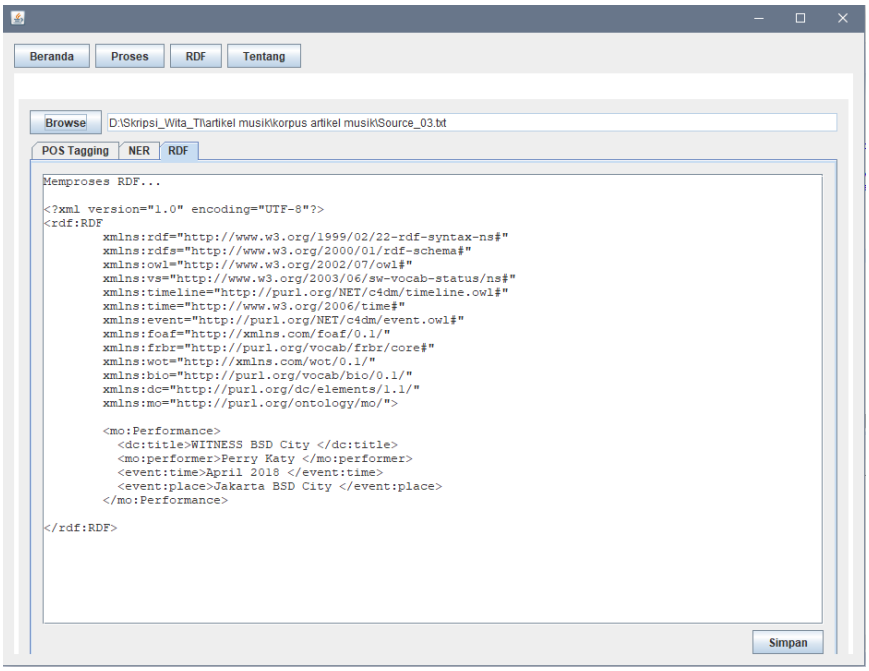
```
<?xml version="1.0" encoding="UTF-8"?>
//a
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:vs="http://www.w3.org/2003/06/sw-vocab-status/ns#"
    xmlns:timeline="http://purl.org/NET/c4dm/timeline.owl#"
    xmlns:time="http://www.w3.org/2006/time#"
    xmlns:event="http://purl.org/NET/c4dm/event.owl#"
    xmlns:foaf="http://xmlns.com/foaf/0.1/"
    xmlns:frbr="http://purl.org/vocab/frbr/core#"
    xmlns:wot="http://xmlns.com/wot/0.1/"
    xmlns:bio="http://purl.org/vocab/bio/0.1/"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:mo="http://purl.org/ontology/mo/">

//b
    <mo:Performance>
      <dc:title> Festival SHVR Ground </dc:title>
      <mo:performer> Khalid </mo:performer>
      <event:time>4 Mei 2018</event:time>
      <event:place> Jakarta </event:place>
    </mo:Performance>

</rdf:RDF>
```

dimana :

a. PREFIX; merupakan *namespace* yang digunakan dalam ontologi musik sebagai *identifier* agar ontologi lebih mudah dipahami.
b. Hasil Ekstraksi; <mo:Performance><Performance>, merupakan *class* yang diambil dari ontologi musik dipakai untuk menandakan bahwa data-data yang diambil merupakan bagian dari suatu *performance*; <dc:title> Festival SHVR

Ground </dc:title>, merupakan *property* yang memberikan identitas ‘Festival SHVR Ground’ sebagai entitas *title* pada rdf; <mo:performer> Khalid </mo:performer>, merupakan *property* yang memberikan identitas ‘Khalid’ sebagai entitas *performer* pada rdf; <event:time>4 Mei 2018</event:time>, merupakan *property* yang memberikan identitas ‘4 Mei 2018’ sebagai entitas *time* pada rdf; <event:place> Jakarta </event:place>, merupakan *property* yang memberikan identitas ‘Jakarta’ sebagai entitas *place*.

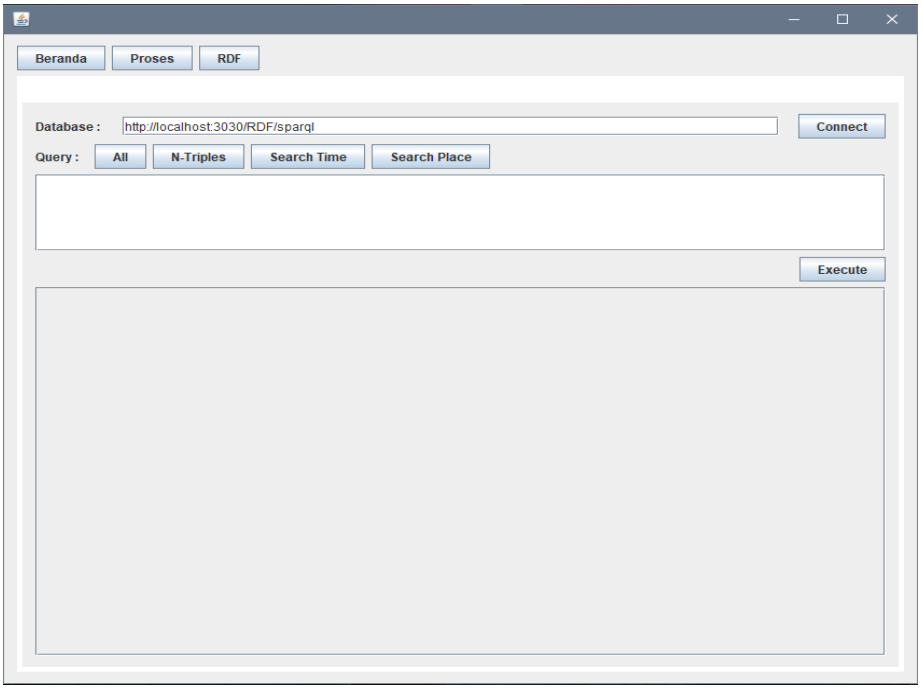
Tampilan hasil ekstraksi informasi bentuk RDF pada aplikasi dapat dilihat pada Gambar 4.5. Pada tampilan ini user dapat menyimpan hasil proses ekstraksi ke dalam bentuk .rdf.



Gambar 4.5. Tampilan Hasil Ekstraksi yang ditampilkan dalam bentuk RDF

3. Halaman RDF

Halaman ini berfungsi untuk menunjukkan hasil ekstraksi informasi yang telah disimpan dalam bentuk .rdf kemudian ditampilkan dengan melakukan koneksi ke dataset yang ada di *SPARQL server* menggunakan *Apache Jena Fuseki* melalui localhost:3030/.

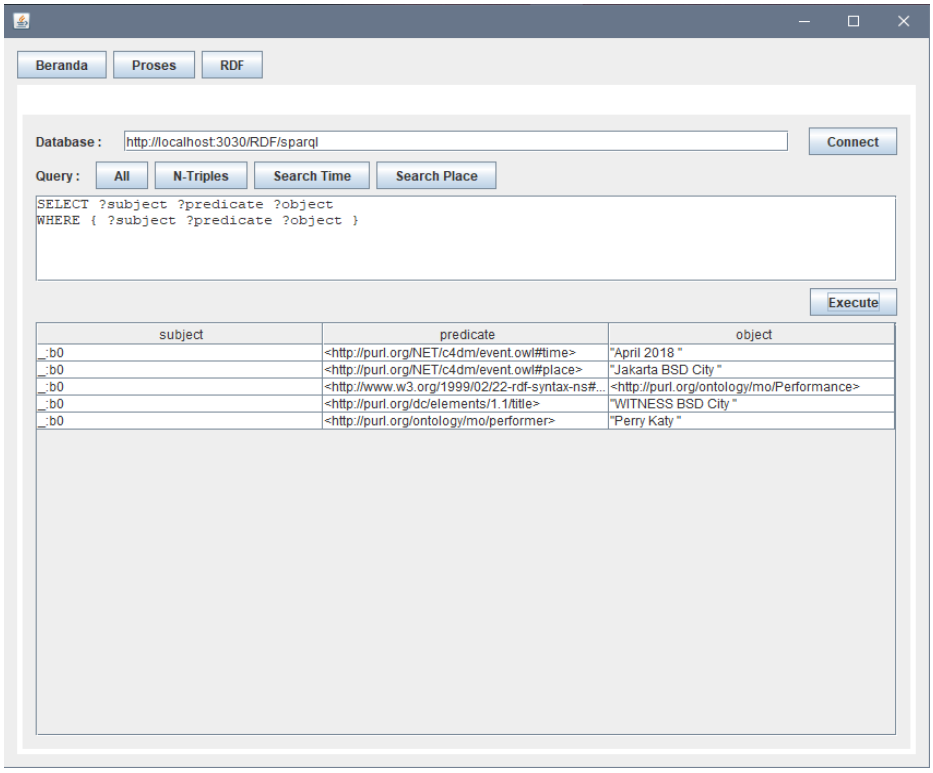


Gambar 4.6. Tampilan Halaman Pemanggilan Output File RDF

Data file RDF yang telah disimpan akan di upload terlebih dahulu ke dalam dataset yang ada di *server Apache Jena Fuseki* agar dapat diambil dan ditampilkan datanya ke dalam sistem. Pada halaman ini terdapat 4 *button* yang masing-masing berisi *SPARQL query*. Untuk memanggil hasil ekstraksi dalam bentuk *n-triples* ke dalam aplikasi digunakan *SPARQL query* yaitu :

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?subject ?predicate ?object
WHERE {
?subject ?predicate ?object
}
```

Dapat di lihat pada Gambar 4.7. hasil dari pemanggilan data rdf hasil ekstraksi informasi menggunakan *SPARQL query*, hasil tersebut ditampilkan dalam format *n-triples* yaitu terdiri dari *subject* yang merupakan *id* dari file yang di upload, *predicate* merupakan *name entity* hasil ekstraksi yang diambil berdasarkan ontologi musik dan *object* yang merupakan *value* atau hasil dari ekstraksi informasi. Hasil tersebut didapat dari 1 file RDF yang telah di upload ke *database server*.

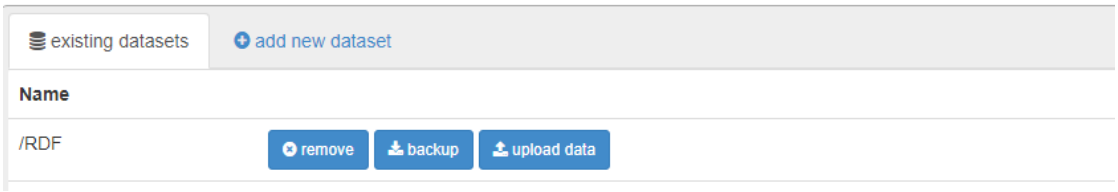


Gambar 4.7. Tampilan Hasil Dari Pemanggilan File RDF

4.2. Pembangunan *Server*

Untuk dapat menunjukkan hasil ekstraksi yang berupa file .rdf menggunakan *SPARQL query*, data-data hasil dari ekstraksi informasi artikel musik yang berupa file .rdf akan disimpan ke dalam *database server*. Dalam implementasinya *file-file* tersebut akan disimpan ke dalam *server* menggunakan *Apache Jena Fuseki*.

Server RDF disimpan pada local *server* (http://localhost:3030/). Gambar 4.8. menunjukkan tampilan pada *server Apache Jena Fuseki* file RDF.



Gambar 4.8. Tampilan pada *Server Apache Jena Fuseki* File RDF

4.3. Pengujian Sistem

Pada tahap ini dilakukan pengujian dengan memanggil beberapa artikel berita seperti sindonews.com, bintang.com, metrotvnews.com, tribunnews.com.

Berikut adalah contoh data yang diambil dari salah satu berita online:

Janji Katy Perry kepada Fans Menjelang Tur WITNESS di Jakarta

Katy Perry akan kembali menyapa para penggemarnya di Indonesia. Kali ini ia akan membawakan banyak hal spesial dalam rangkaian tur "WITNESS"-nya.

Dalam tur kali ini, Katy Perry akan tampil spesial di hadapan para penggemarnya. Katy Perry mengumumkan dirinya akan membawa semua perlengkapan dan segala pernak-pernik show-nya ke Asia.

Ini akan menjadi aksi yang sama seperti konser Katy Perry sebelumnya di Amerika Utara. Katy Perry ingin tampil istimewa dalam tur "WITNESS"-nya di Indonesia nanti.

Katy Perry melakukannya agar semua penggemar bisa melihat, menyaksikan, mendengarkan, dan merasakan secara langsung betapa istimewanya konser kali ini.

Katy Perry akan hadir bersama tim yang hebat dan meyakinkan bahwa turnya kali ini memiliki skala produksi terbesar. Tur "WITNESS" ini diramalkan bakal menjadi pertunjukan konser yang paling heboh.

Dalam konsernya kali ini, Katy Perry membawa 50 ton kargo perlengkapan konsernya. Katy Perry juga akan berganti kostum enam kali dan membawa baju khusus untuk setiap show-nya.

Katy Perry juga akan membawakan 11 lagu hitsnya dengan 10 penari dan juga dua penyanyi latar. Dalam konsernya kali ini, ia juga membawa lima pemain musiknya, termasuk pemain basnya yang telah bersama Katy sejak album pertama.

Dalam tur "WITNESS" ini juga dikabarkan Katy Perry kemungkinan besar akan membawa ikan hiu untuk berada di antara para penggemarnya.

Konser tur “WITNESS” Katy Perry di Jakarta ini akan diselenggarakan pada 14 April 2018 di BSD City.

Setelah input teks dimasukkan, maka sistem akan mulai melakukan ekstraksi informasi sesuai proses dan aturan-aturan yang telah ditentukan. Setelah selesai dilakukan ekstraksi informasi, hasil dari ekstraksi informasi pada awalnya akan ditampilkan dalam bentuk RDF(.rdf) pada aplikasi. Hasil ekstraksi informasi yang berupa file RDF ini akan di simpan dan di upload ke dalam *SPARQL server* lalu dimasukkan ke dalam satu *database* RDF untuk kemudian ditampilkan dengan baik ke dalam aplikasi. Contoh f*ile* RDF yang dihasilkan, yaitu:

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:vs="http://www.w3.org/2003/06/sw-vocab-status/ns#"
    xmlns:timeline="http://purl.org/NET/c4dm/timeline.owl#"
    xmlns:time="http://www.w3.org/2006/time#"
    xmlns:event="http://purl.org/NET/c4dm/event.owl#"
    xmlns:foaf="http://xmlns.com/foaf/0.1/"
    xmlns:frbr="http://purl.org/vocab/frbr/core#"
    xmlns:wot="http://xmlns.com/wot/0.1/"
    xmlns:bio="http://purl.org/vocab/bio/0.1/"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:mo="http://purl.org/ontology/mo/">

    <mo:Performance>
      <dc:title>WITNESS BSD City </dc:title>
      <mo:performer>Perry Katy </mo:performer>
      <event:time>April 2018 </event:time>
      <event:place>Jakarta BSD City </event:place>
    </mo:Performance>

</rdf:RDF>
```

Setelah file tersebut di upload maka untuk memanggil data yang ada pada file RDF tersebut dibutuhkan *SPARQL query*, contoh *query* yang digunakan untuk menampilkan semua hasil ekstraksi adalah seperti dibawah ini:

```
//a
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

//b
SELECT ?subject ?predicate ?object
WHERE {
?subject ?predicate ?object
}
```

dimana:

a. PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>; merupakan *namespace* yang digunakan pada *query* ini
b. SELECT ?subject ?predicate ?object WHERE { ?subject ?predicate ?object }; merupakan *query* untuk menampilkan hasil dari ekstraksi yang berupa *n-triples* yaitu *subject* (ID), *predicate* (*name entity*), *object* (*value*).

Hasil pengambilan data ekstraksi informasi artikel musik melalui *query* tersebut yaitu:

Tabel 4.1. Hasil Ekstraksi Informasi Artikel Musik

Subject	***Predicate***	***Object***
_:b01	<http://purl.org/NET/c4dm/event.owl#time>	April 2018
_:b01	<http://purl.org/NET/c4dm/event.owl#place>	Jakarta BSD City
_:b01	<http://purl.org/dc/elements/1.1/title>	WITNESS BSD City
_:b01	<http://purl.org/ontology/mo/performer>	Perry Katy

Setelah dilakukan pemeriksaan artikel oleh sistem, maka didapatlah hasil sesuai yang ditunjukkan pada Tabel 4.1. Nama yang kurang tepat dan tanggal pada bagian waktu tidak terdeteksi oleh sistem dikarenakan proses NER kurang bekerja dengan baik dalam melakukan pendeteksian nama orang dan entitas dari angka yang terdapat di depan

entitas TIME seperti ’14 April 2018’. Untuk mengetahui tingkat kebenaran sistem, maka dapat dilakukan perhitungan nilai *f-score* dari tiap-tiap entitas ekstraksi informasi yang diambil dengan terlebih dahulu harus mengetahui nilai *true positive* (TP), *false positive* (TN), *false negative* (TP), *precision* dan *recall*. Berikut hasil perhitungan nilai *f-score* yang didapat dari hasil pengujian 50 artikel berita musik yang berhubungan dengan konser dapat dilihat pada Tabel 4.2 berikut

Tabel 4.2. Nilai F-Score Hasil Pengujian Sistem

	TP	FP	FN	*Precision*	*Recall*	*F-Score*
Performer (Artis)	107	42	32	71,81%	76,97%	74,30%
Time (Waktu)	114	0	28	100%	80,28%	89,06%
Place (Tempat)	137	18	21	88,38%	86,70%	87,54%
Title (Acara)	108	46	31	70,13%	77,69%	73,72%

Dengan melakukan pengujian sistem ekstraksi informasi pada artikel musik yang berhubungan dengan penyelenggaraan acara musik atau konser menggunakan 50 data artikel dari web berita seperti sindonews.com, bintang.com, metrotvnews.com, tribunnews.com maka diperoleh nilai *f-score* dari masing-masing entitas yang diambil. Hasil menunjukkan bahwa nilai *f-score* dari entitas *time* yaitu 89,06% dan *place* yaitu 87,54% lebih tinggi dari nilai *f-score* entitas *performer* yaitu 74,30% dan *title* yaitu 73,72%. Hasil ini di dapat karena NER tidak bekerja dengan baik dalam melakukan pendeteksian nama dan juga dalam pendeteksian organisasi.

Adapun kelemahan ekstraksi informasi artikel musik dengan menggunakan NER dan metode *rule base* ini adalah sistem hanya bisa mengenali suatu kata yang merupakan entitas nama, lokasi, dan organisasi jika input *case* huruf sudah benar. Sistem hanya dapat mendeteksi entitas yang dimiliki jika didahului oleh *list-list* kata tertentu, salah satu contoh seperti sistem tidak dapat mengenali tanggal pada entitas *date* dan yang terdeteksi hanya bulan atau tahun. Sistem juga mengalami kesulitan dalam mendeteksi singkatan-singkatan pada teks dikarenakan sistem tidak bisa mengenali apakah kata tersebut merupakan singkatan atau tidak.

BAB V

KESIMPULAN DAN SARAN

Bab ini membahas tentang kesimpulan dari penerapan metode yang diajukan untuk melakukan sistem ekstraksi informasi artikel musik berdasarkan ontologi serta saran-saran dalam pengembangan yang dapat digunakan untuk penelitian selanjutnya.

5.1. Kesimpulan

Berdasarkan pengujian dan penelitian yang dilakukan mengenai ekstraksi informasi artikel musik berdasarkan ontologi menggunakan NER dengan metode *rule based*, adapun kesimpulan yang dapat diambil yaitu:

1. Proses pengenalan kelas kata dan entitas kata pada data uji menjadi salah satu penyebab berkurangnya performa sistem untuk mengambil data dalam proses ekstraksi karena adanya kesalahan dalam hasil pengenalannya yang disebabkan input teks tidak memiliki struktur kalimat yang benar dan juga input teks yang tidak memiliki *case* huruf yang benar.
2. Gagalnya sistem untuk mengekstraksi kata yang tepat disebabkan kurang maksimalnya *rule-rule* yang dibuat dan diterapkan.

5.2. Saran

Adapun saran untuk pengembangan penelitian selanjutnya adalah:

1. Diharapkan penelitian selanjutnya dapat menggunakan metode yang lain pada proses ekstraksi informasi sehingga dapat dibandingkan dengan menggunakan metode *rule based*.
2. Diharapkan pada penelitian selanjutnya dapat menggunakan data pelatihan atau korpus yang baik dan banyak pada proses *pre-processing* sehingga pada proses selanjutnya sistem dapat menghasilkan akurasi yang lebih tinggi.
3. Diharapkan pada penelitian selanjutnya *rule-rule* yang ada diperbanyak untuk menyesuaikan pada jenis artikel musik yang tidak hanya sebatas acara musik.

DAFTAR PUSTAKA

Charisma, I. 2016. *Clustering* Artikel Web Kesehatan Dengan Menggunakan Algoritma *Self Organizing Maps*. Skripsi. Universitas Sumatera Utara.

Evans, L.C. 1999. *An Introduction to Stochastic Differential Equations Version 1.2*. U.C.Berkeley: Departemen of Mathematics.

Fachri, M. 2014. *Pengenalan Entitas Bernama pada Teks Bahasa Indonesia Menggunakan Hidden Markov Model*. Yogyakarta. Skripsi.

Hayadi, H.B. 2016. *Sistem Pakar*. Yogyakarta: Deepublish.

Horridge, M. 2011. *A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools Edition 1.3*. The University of Manchester: England.

Ilyas, R., & Khodra, M.L,. 2015. Ekstraksi Informasi 5W1H pada Berita Online Bahasa Indonesia. *Jurnal Cybermatika* 3(1): 35-41.

Ismaya, Agny. 2014. Algoritma Ekstraksi Informasi Berbasis Aturan. *JNTETI* 3(4): 242-247.

Jiang, J. 2012. *Information Extraction from Text. In Mining Text Data*, edited by C. C. Aggarwal and C. Zhai. pp 11 – 14. Springer: New York.

Jurafsky, D S. 2000. Speech and Language Processing "An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Prentice-Hall, Inc. New Jersey.

Kamus Besar Bahasa Indonesia. http://www.pusatbahasa.diknas.go.id, access date 12 April 2010.

Mooney, U. Y. N. a. R. J. 2002. *Text Mining with Information Extraction*, Austin: s.n.

Nandini, D. 2014. *Semantic Web and Ontology (1st edition)*. Bookboon: Denmark.

Noy, N.F. & McGuinness, D.L. 2001. Ontology Development 101: A Guide to Creating Your First Ontology. *Knowledge Systems Laboratory (KSL) of Department of Computer Science Stanford*. USA: Technical Report, KSL-01-05.

Permadi, A.D. 2009. Ekstraksi Data pada Halaman Web Menggunakan *Partial Tree Alignment (DEPTA)*. Skripsi. Universitas Telkom.

Pisceldo, F., Adriani, M. & Manurung, R. 2009. Probabilistic Part Of Speech Tagging for Bahasa Indonesia. *Proceeding in Third International MALINDO Workshop*.

Putra, J.W.G. 2018. *Pengenalan Konsep Pembelajaran Mesin dan Deep Learning.*

Rabiner, L. 1989. *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition.* New Jersey: Prentice Hall.

Riaddy, A.I. 2016. Ekstraksi Informasi pada Makalah Ilmiah dengan Pendekatan *Supervised Learning*. Skripsi. Universitas Telkom.

Setiyoaji, A., Muflikhah, L. & Fauzi, M.A. 2017. Named entity recognition menggunakan hidden markov model dan algoritma viterbi pada teks tanaman obat. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* 1(12):1858-1864.

Susanti, E. 2015. Ekstraksi Informasi Konten Web Menggunakan Pendekatan Berbasi Ontologi. *Jurnal Teknologi Technoscientia*. 7(2) :128-136.

Tala, F.Z. 2003. *A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia*. Tesis. Institute for Logic, Languange and Computation. Universiteti van Amsterdam the Netherlands.

Wibisono, Y. 2018. Penggunaan *Hidden Markov Model* untuk kompresi kalimat. Tesis. Institut Teknologi Bandung.

Wicaksono, A.F. & Purwarianti, A. 2010. HMM based Part Of Speech Tagger for Bahasa Indonesia . *Proceeding of 4th International Malindo (Malay and Indonesian Language) Workshop*, Jakarta.

Wood, D., Zaidman, M., Ruth, L. & Hausenblas, M. 2014. *Linked Data: Structured Data on the Web 1st Edition*. Manning Publications Co.: Shelter Island.

KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
UNIVERSITAS SUMATERA UTARA
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
Jalan Universitas No. 9A Kampus USU, Medan-20155
Telepon/Fax : 061-8228048 Laman: http://fasilkom-ti.usu.ac.id

No. : 656 /UN5.2.1.14/LLS/2019 22 April 2019
Hal : Kesediaan Menguji

Yth. Sdr.....
Tenaga Pendidik Fasilkom-TI USU
Medan

Dengan hormat, bersama ini diharapkan kesediaan Saudara untuk menguji Tugas Akhir (Ujian Komprehensif) dari mahasiswa:

Nama : Juwita Shifa Rahmah
NIM : 131402111
Program Studi : S1 Teknologi Informasi

Yang akan dilaksanakan pada :

Hari / Tanggal : Kamis / 25 April 2019
Pukul : 09.00 WIB s/d selesai
Tempat : Ruang Seminar S-1 Program Studi Teknologi Informasi Fasilkom-TI USU

Demikian disampaikan, atas kehadiran Saudara tepat pada waktunya diucapkan terima kasih.

Wakil Dekan I

Dr. Elviawaty Muisa Zamzami, ST,MT,MM
NIP. 19700716 200501 2 002

Tembusan:
1. Dekan Fasilkom-TI USU
2. Ketua Program Studi S-1 Teknologi Informasi Fasilkom-TI USU
3. Bendahara Pembantu Pengeluaran Fasilkom-TI USU
4. Arsip

KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
UNIVERSITAS SUMATERA UTARA
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
Jalan Universitas No. 9A Kampus USU Medan 20155
Telepon/Fax : 061-8228048 Laman: http://fasilkomti.usu.ac.id

KEPUTUSAN
DEKAN FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
NOMOR: 672 /UN5.2.1.14/SK/LLS/2019

Dekan Fakultas Ilmu Komputer dan Teknologi Informasi USU

Membaca : Surat permohonan mahasiswa Fasilkom-TI USU tanggal 7 Januari 2019 perihal permohonan ujian skripsi:

Nama	: Juwita Shifa Rahmah
NIM	: 131402111
Program Studi	: S-1 Teknologi Informasi
Judul Skripsi	: Ekstraksi Informasi Artikel Musik Berdasarkan Ontologi Menggunakan Named Entity Recognition dan Metode Rule Based

Memperhatikan : Bahwa mahasiswa tersebut telah memenuhi kewajiban dan persyaratan untuk ikut dalam pelaksanaan ujian skripsi mahasiswa pada Program Studi S-1 Teknologi Informasi Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara T.A. 2017/2018.

Menimbang : Bahwa permohonan tersebut di atas dapat disetujui dan perlu ditetapkan dengan surat keputusan.

Mengingat :
1. Undang-undang Nomor 20 Tahun 2003 tentang Sistem Pendidikan Nasional.
2. Peraturan Pemerintah Nomor 17 tahun 2010 tentang pengelolaan dan penyelenggara pendidikan.
3. Keputusan Rektor USU Nomor 701/UN5.1.R/SK/SPB/2013 tentang Peraturan Akademik Program Sarjana Universitas Sumatera Utara.
4. Surat Keputusan Rektor USU Nomor 832/UN5.1.R/SK/SDM/2016 Tanggal 16 Mei 2016 tentang pengangkatan Dekan Fasilkom-TI USU Periode 2016-2021.

MEMUTUSKAN

Menetapkan
Pertama : Membentuk dan mengangkat Tim Penguji Skripsi mahasiswa sebagai berikut:

Ketua	: Prof. Dr.Opim Salim Sitompul,M.Sc NIP. 19610817 198701 1 001
Sekretaris	: Ainul Hizriadi,S.Kom,M.Sc NIP. 19851027 201706 1 001
Anggota Penguji	: Dani Gunawan,ST,MT NIP. 19820915 201212 1 002
Anggota Penguji	: Dedy Arisandi,ST.M.Kom NIP. 19790831 200912 1 002
Moderator	: -
Panitera	: -

Kedua : Segala biaya yang diperlukan untuk pelaksanaan kegiatan ini dibebankan pada Dana Penerimaan Bukan Pajak (PNPB) Fasilkom-TI USU Tahun 2018.

Ketiga : Keputusan ini berlaku sejak tanggal ditetapkan dengan ketentuan bahwa segala sesuatunya akan diperbaiki sebagaimana mestinya apabila dikemudian hari terdapat kekeliruan dalam surat keputusan ini.

Ditetapkan di Medan
Pada Tanggal: 22 April 2019
Dekan,

Universitas Sumatera Utara

Opim Salim Sitompul
NIP. 19610817 198701 1 001

Tembusan:
1. Ketua Program Studi S-1 Teknologi Informasi