Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

Team project ©2017
Dony Pratidana S. Hum | Bima Agus Setyawan S. IIP

BAB III
METODOLOGI DAN PERANCANGAN SISTEM

3.1. Analisis Kebutuhan

Untuk melakukan penelitian dibutuhkan perangkat keras dan piranti lunak. Rincian mengenai perangkat keras dan piranti lunak yang dibutuhkan sebagai berikut.

3.1.1. Perangkat Keras yang Digunakan

Perangkat keras yang digunakan sebagai berikut:

1. **Komputer dengan spesifikasi**
   a. Intel Core 2 Duo T6600 @ 2.20 Ghz
   b. RAM 3 Gb
   c. Windows 7 32-bit
2. **Server dengan spesifikasi**
   a. Intel Xeon
   b. RAM 24 Gb
   c. Koneksi internet milik *idhostinger.com*

3.1.2. Piranti Lunak yang Digunakan

Piranti lunak yang digunakan sebagai berikut:

1. Webserver Apache 1.7.4

Implementasi Ontologi ..., Alexander tyas Aji Wardhana, FTI UMN, 2013

2. PHP 5.3.5
3. MySQL 5.1.
4. Mozilla Firefox 21.0
5. Google Chroome 27.0.
6. Firebug 1.11.4
7. Protégé 3.4.8
8. Protégé 4.3.0
9. Glify Online Diagram

3.2. Metodologi Penelitian

3.2.1. Studi Literatur

Penelitian diawali dengan melakukan studi literatur menggunakan referensi buku, jurnal, dan artikel mengenai web semantik, Ontologi Friend of a Friend (FOAF) dan Ontologi Musik. Literatur tersebut dijadikan pedoman dalam melakukan penelitian.

3.2.2. Perancangan Sistem

Sistem dirancang sesuai dengan pendekatan *System Development Life Cycle* (Valacich, 2012) untuk membangun rancangan sistem dengan menggunakan siklus yang spesifik sehingga sistem dapat mendukung kebutuhan pengguna yaitu untuk mendapatkan informasi konser musik yang sesuai. Siklus ini akan terus berputar agar sistem dapat beradaptasi dengan karakteristik masing-masing pada seluruh sumber data dan pada akhirnya dapat mencapai tujuan untuk

merekomandasikan konser musik kepada pengguna aplikasi Facebook. Berikut ini merupakan tahap-tahap dalam pembangunan sistem rekomendasi konser musik berdasarkan SDLC.

1. Perencanaan

Hasil observasi telah dijelaskan dalam latar belakang bahwa sistem rekomendasi konser musik dengan menggunakan Ontologi Friend of A Friend (FOAF) dan Ontologi Musik pada *platform* Facebook perlu dikembangkan pada aplikasi rekomendasi konser musik. Dengan melakukan implementasi ontologi pada aplikasi sistem rekomendasi konser musik, maka hirarki struktur data yang digunakan menjadi jelas.

Sistem dirancang agar dapat beradaptasi terhadap seluruh data yang diperoleh dari berbagai sumber dengan karakteristik API yang berbeda-beda dan dapat menampilkan *output* berupa konser musik sesuai dengan profil akun Facebook pengguna. Terdapat tiga fitur utama, yaitu untuk menampilkan konser musik artis favorit, konser musik di lokasi *user*, dan konser musik dari *similar artist*.

2. Analisis

Untuk mendapatkan personal data pengguna Facebook yang akan menggunakan sistem rekomendasi konser musik digunakan Facebook Graph API dan Facebook SDK untuk bahasa pemrograman PHP.

Data konser musik diperoleh dari Songkick dan Eventbrite dan akan diakses melalui API yang disediakan yang kemudian akan direspon melalui JSON. Data dalam bentuk JSON tersebut akan melalui proses *decode* sehingga

dapat digunakan menggunakan bahasa pemrograman PHP. API Songkick dan API Eventbrite memiliki karakteristik yang berbeda. Songkick menyediakan API yang memungkinkan untuk melakukan *request* informasi konser musik dengan melakukan *passing* nama artis atau lokasi. Untuk parameter lokasi, Songkick meminta *input* dengan format *geocode* sehingga dibutuhkan API Google Maps untuk konversi nama lokasi menjadi format *latitude* dan *longitude*. Sedangkan Eventbrite memungkinkan *request* data konser musik dengan melakukan *passing* nama kota lokasi.

Untuk mendapatkan *similar artist*, digunakan DBpedia sebagai penyedia informasi terstruktur dengan melakukan *passing query* SPARQL dan hasilnya di-*grab* menggunakan PHP Client URL Library (CURL).

Seluruh data tersebut akan diinput ke dalam *triplestore* dengan menggunakan SPARQL *query* menggunakan *framework* ARC2. Untuk memenuhi kebutuhan aplikasi sistem rekomendasi konser musik, dilakukan pembuatan *custom property* dari *class* yang telah didefiniskan pada *vocabulary* Ontologi FOAF maupun Ontologi Musik.

3. Perancangan

Berdasarkan *requirement* yang telah didefinisikan pada tahap perancanaan dan tahap analisis, fase perancangan dimulai dengan mendefinisikan seluruh kebutuhan sistem dan merancang sistem dengan menggunakan diagram arsitektur, membuat diagram alir, dan melakukan pemodelan implementasi Ontologi FOAF dan Ontologi Musik menggunakan Protégé 3.4.8, dan merancang

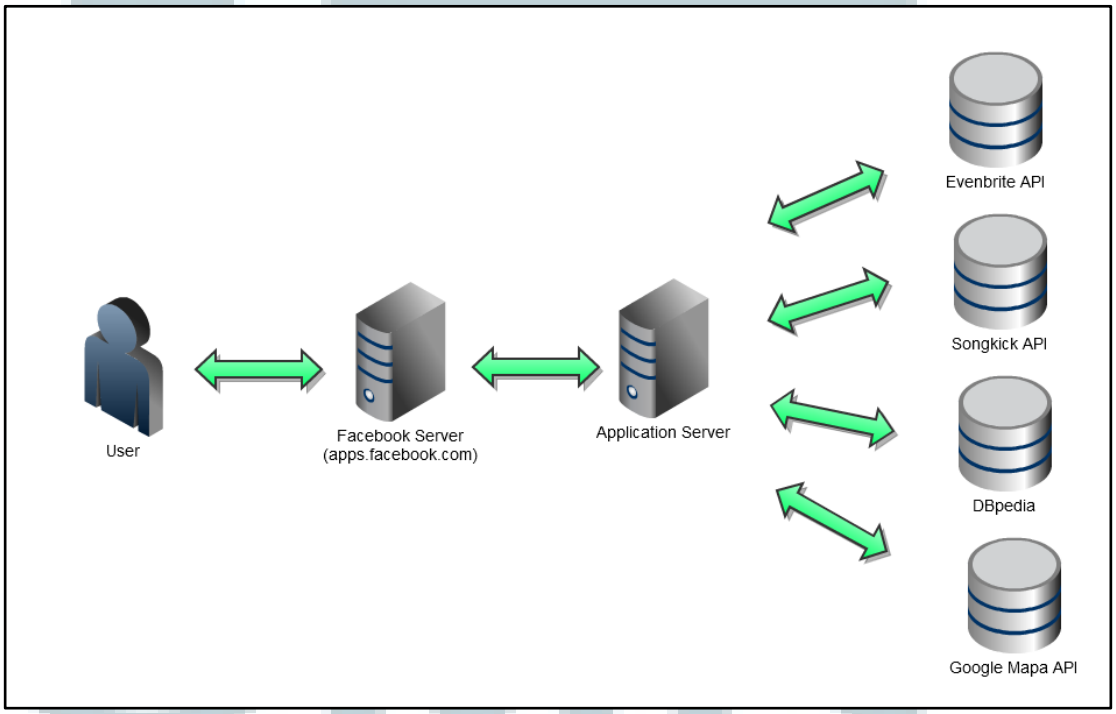
tampilan antar muka aplikasi. Hasil keseluruhan rancangan ontologi ditampilkan pada lampiran.

4. Implementasi

Fase implementasi adalah proses implementasi sistem dan pengujian sistem berdasarkan fungsionalitasnya dan kesesuaian *output* sistem rekomendasi konser musik.

3.2.2.1. Diagram Arsitektur Sistem

Diagram Arsitektur Sistem menjelaskan arsitektur sistem secara umum. Diagram Arsitektur Sistem digambarkan oleh Gambar 3.1.



Gambar 3.1. Diagram Arsitektur Sistem

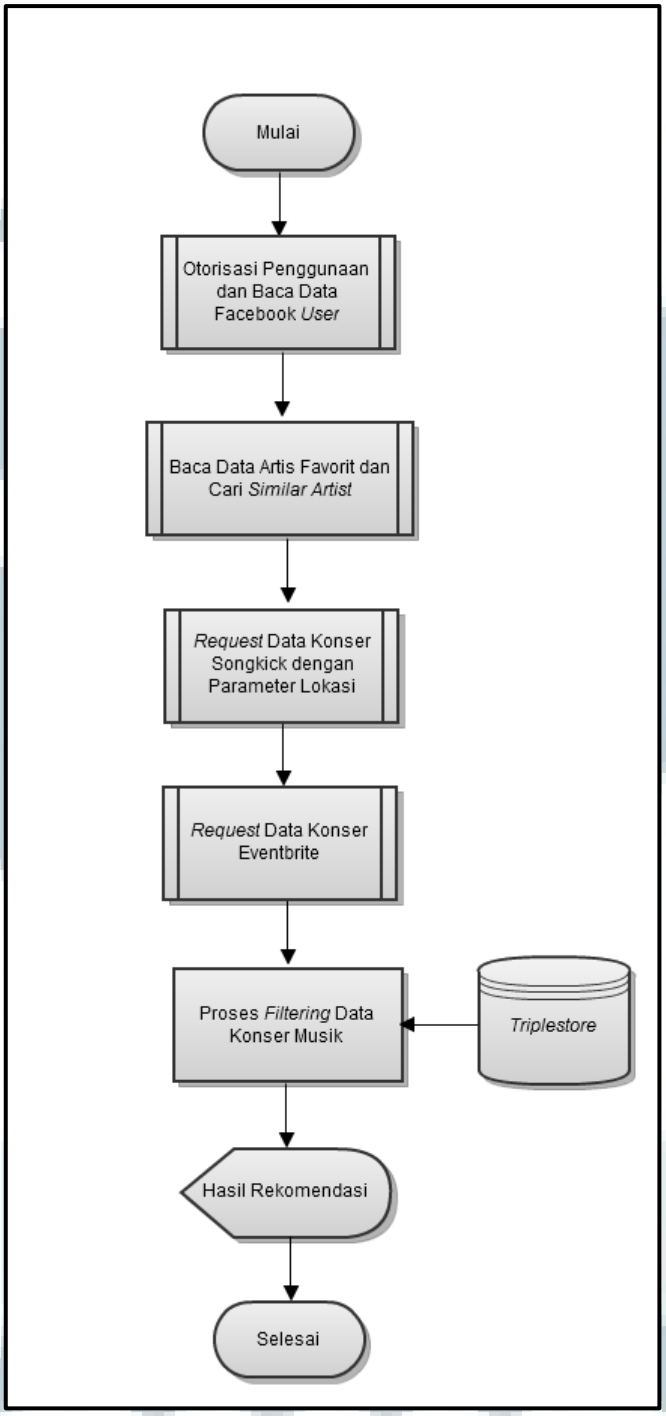
Ketika aplikasi sistem rekomendasi konser musik diakses oleh *user* melalui web *browser*, *user* akan melakukan *request* kepada server Facebook kemudian server Facebook akan menyampaikan *user request* tersebut menuju server aplikasi. Server aplikasi akan menyusun API Facebook agar bisa diunggah ke dalam sebuah kanvas kosong yang kemudian kanvas tersebut akan memuat tampilan aplikasi dengan memanggil alamat URL yang sudah didaftarkan pada saat pembuatan aplikasi Facebook.

Data konser musik diambil dari server Songkick dan Evenbrite melalaui API dalam format JSON, kemudian untuk mendapatkan *similar artist* digunakan *knowledge base* dari DBpedia berdasarkan genre yang sama.

3.2.2.2. Diagram Alir

Berdasarkan diagram umum pada Gambar 3.1, diagram alir dibagi menjadi lima bagian, yaitu diagram alir umum, diagram alir untuk menggambarkan proses otorisasi akses data pengguna Facebook dan membaca data, diagram alir untuk menggambarkan proses input data artis favorit dan *similar artist*, diagram alir untuk menggambarkan pengambilan data konser musik dari Songkick menggunakan parameter nama artis dan lokasi, dan diagram alir untuk pengambilan data konser musik dari Eventbrite.

1. Diagram Alir Umum



Gambar 3.2. Diagram Alir Umum

Gambar 3.2 menggambarkan diagram alir secara umum proses yang terjadi pada sistem rekomendasi konser musik. Diagram alir umum hanya

menampilkan alir sistem secara umum. Terdapat fase *predefined process* yang berarti rincian proses dijelasaskan pada diagram alir lain. Rincian diagram alir proses otorisasi pengguna Facebook dan baca data pengguna Facebook ditampilkan pada Gambar 3.3, proses baca data artis favorit dan pencarian *similar artist* ditampilkan pada Gambar 3.4, proses *request* data konser musik Songkick dengan parameter lokasi ditampilkan pada Gambar 3.7, dan proses *request* data konser Evenbrite menggunakan parameter nama lokasi ditampilkan pada Gambar 3.8.

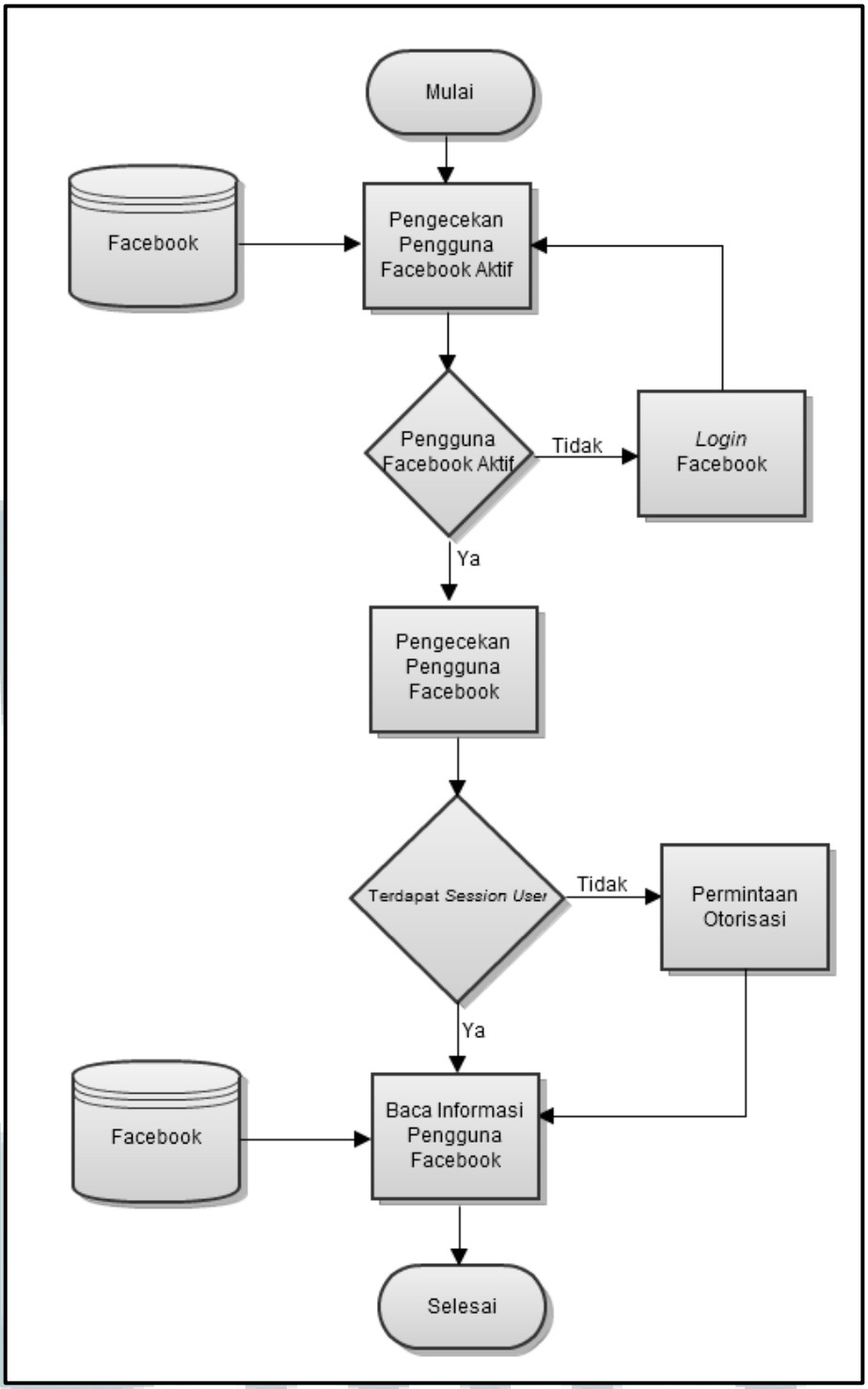
Pada dasarnya diagram alir tersebut menjelaskan proses yang dimulai dengan membaca data pengguna yang diperoleh melalui Facebook Graph API kemudian melakukan *request* data konser musik melalui API Songkick dan API Eventbrite dengan mengirim parameter yang dibutuhkan. Untuk mendapatkan daftar konser musik melalui Songkick, dibutuhkan *passing* parameter lokasi konser musik dengan format *geocode*, sehingga dibutuhkan Google Maps API untuk konversi nama lokasi menjadi format *geocode*.

Seluruh data yang diperoleh akan diinput ke dalam *triplestore* dengan menggunakan SPARQL *query* dan akan melalui proses *filtering* berdasarkan data yang ada pada akun Facebook untuk menampilkan *output* berupa informasi konser musik.

2. Diagram Alir Otorisasi Penggunaan dan Baca Data Facebook User

Untuk mendapatkan informasi personal data pengguna Facebook dibutuhkan persetujuan otorisasi oleh *user* untuk menggunakan data-data tersebut hingga data-data tersebut dapat diolah menjadi suatu sistem rekomendasi.

UMN



Gambar 3.3. Diagram Alir Otorisasi dan Penarikan Data Pengguna Facebook

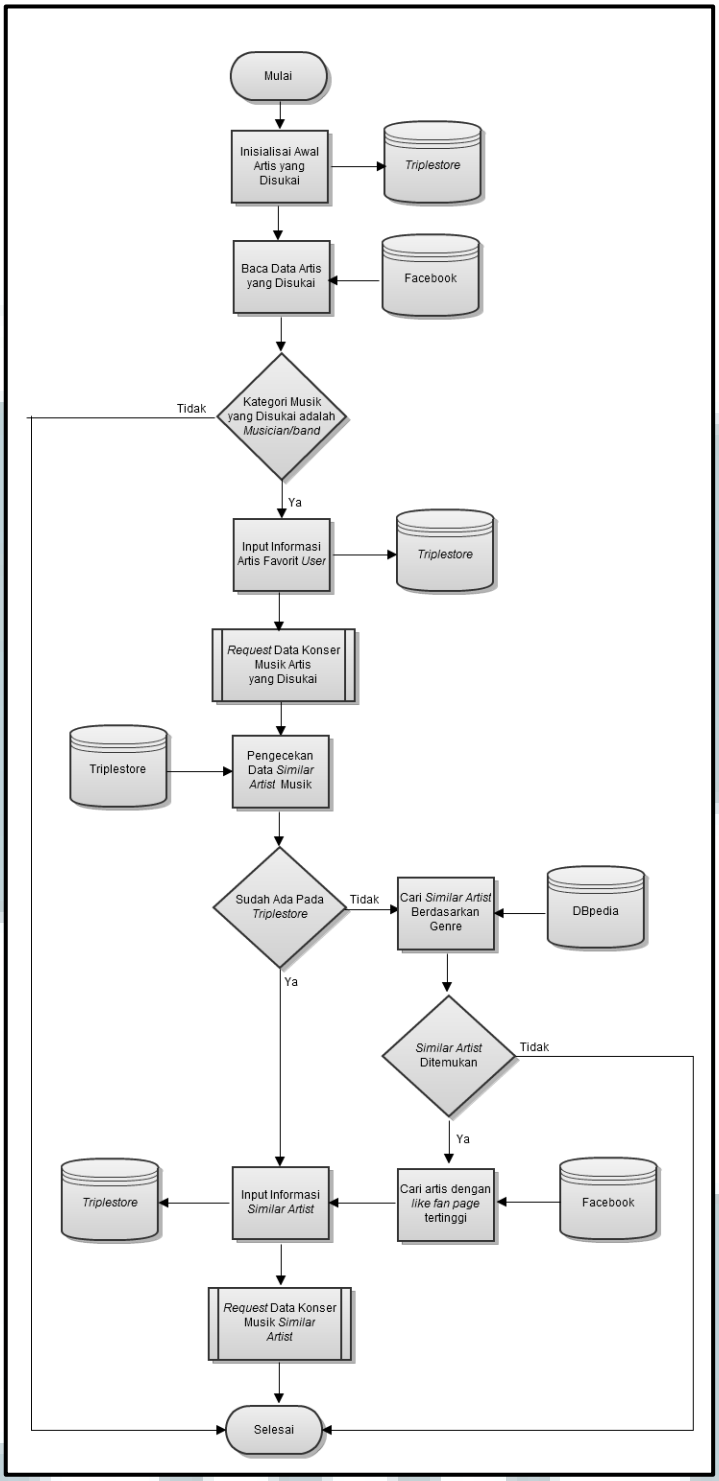
Gambar 3.3 menggambarkan proses permintaan otorisasi sistem untuk menggunakan *personal data* pengguna Facebook. Seorang pengguna sistem

rekomendasi harus terlebih dahulu memiliki akun Facebook dan bersedia untuk memberikan otorisasi akses data kepada sistem. Untuk penggunaan aplikasi kedua kali dan seterusnya, pengguna tidak akan diminta lagi untuk memberikan otorisasi akses data karena sudah terdapat *session* yang menunjukkan bahwa *user* tersebut aktif pada aplikasi tersebut, sehingga data akan langsung bisa digunakan untuk kebutuhan sistem rekomendasi konser musik.

2. Diagram Baca Data Artis Favorit dan Cari *Similar Artist*

Untuk mendapatkan informasi konser musik artis favorit dan rekomendasi, diperlukan data artis musik yeng telah di-*like*. Untuk pencarian *similar artis* dari artis yang telah di-like tersebut digunakan *knowledge base* Dbpedia.

Untuk mendapatkan konser musik dari setiap artis favorit dan *similar artist* yang diperoleh, dilakukan HTTP *request* melalui API Songkick dengan mem-*passing* nama artis tersebut.



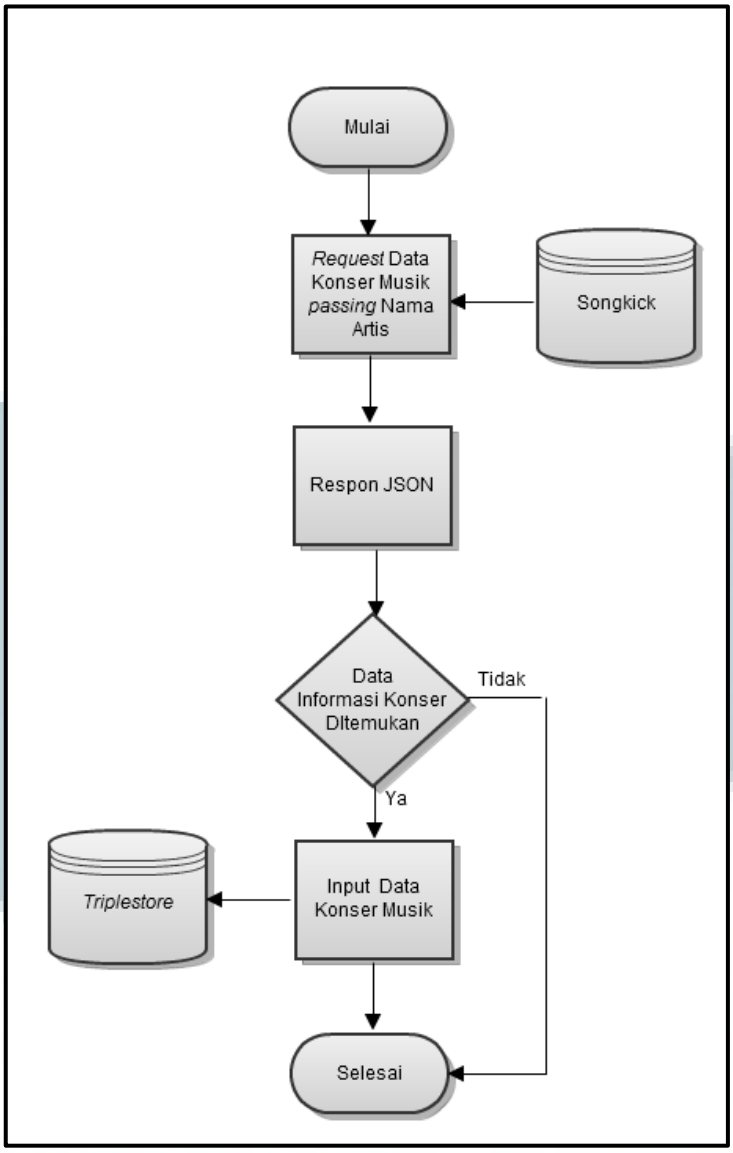
Gambar 3.4. Diagram Alir Input Data Musik yang Disukai oleh Pengguna Facebook

Gambar 3.4 menggambarkan proses dari proses inisialisasi awal hingga input data konser musik dari artis favorit dan *similar artist* yang diperoleh. Perlu dilakukan inisialisasi awal data artis musik yang disukai agar data pada *triplestore* selalu terupdate dengan data yang ada pada Facebook, dengan cara menghapus relasi foaf:likedArtist antar subyek sebagai foaf:Person dan obyek nama artis. Hal ini dilakukan karena data artis yang telah di-*like* akan memiliki kemungkinan adanya perubahan, kemudian dilakukan pengecekan apakah kategori artis yang disukai merupakan "musician/band".

Setelah data musik yang di-*like* diperoleh, dilakukan input data ke dalam *triplestore* kemudian dilakukan HTTP *request* dengan mem-*passing* nama artis tersebut untuk mendapatkan data informasi konser musik melalui API Songkick.

Untuk mempercepat proses pencarian *similar artist*, dilakukan pengecekan apakah artis tersebut sudah ada di dalam *triplestore*. Jika tidak ditemukan, maka dilakukan proses pencarian *similar artist* di dalam *knowledge base* DBpedia. DBpedia akan mengembalikan dua nama artis yang memiliki genre sama, kemudian untuk memperoleh artis yang lebih populer, kedua artis tersebut dibandingkan jumlah *likes* Facebook *fanpage* mereka. Langkah terakhir adalah melakukan input data yang sudah ada ke dalam *triplestore* melalui SPARQL *query*.

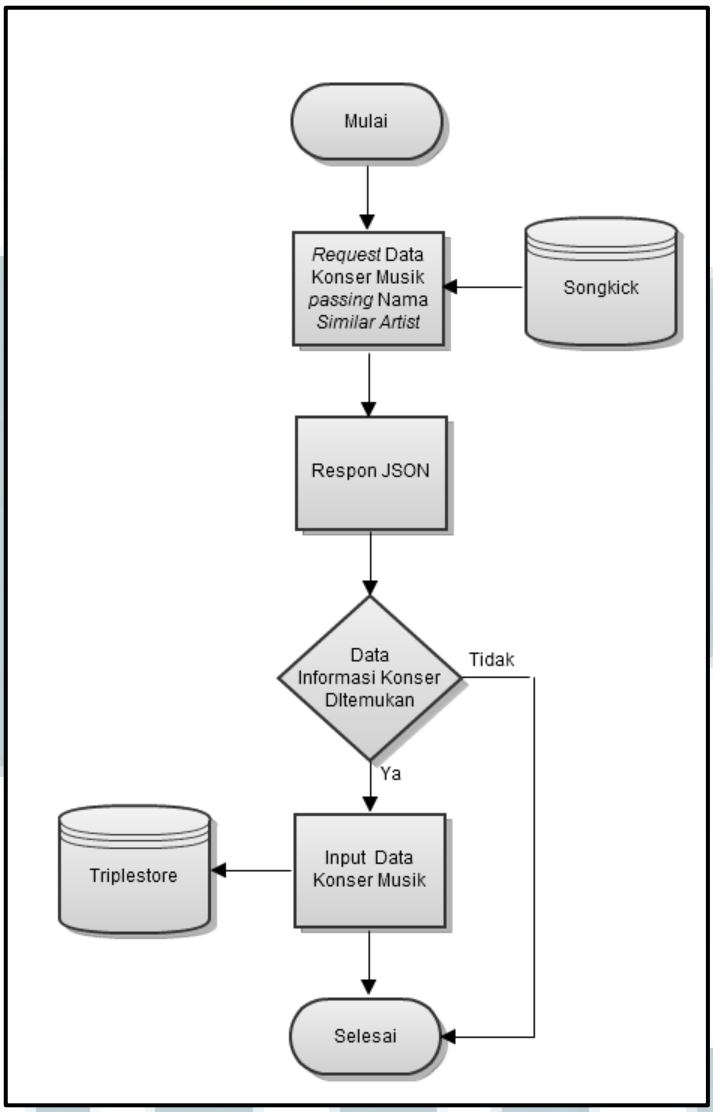
Dalam diagram alir umum seperti pada Gambar 3.4 terdapat *predifined proccess*, yaitu pada proses *request* data konser musik artis favorit dan *similar artist*. Rincian proses *request* data konser musik melalui API Songkick untuk mendapatkan data konser musik artis favorit dijelaskan pada diagram alir berikut.



Gambar 3.5. Diagram *Request* Data Konser Musik Artis yang Disukai

Gambar 3.5 menggambarkan Diagram Alir untuk memperoleh konser musik artis yang disukai dengan melakukan *passing* nama artis dan hasil *request* yang diperoleh akan dikembalikan dalam format JSON. Jika data konser musik ditemukan maka seluruh data konser musik tersebut akan diinput ke dalam *triplestore* menggunakan SPARQL query. Berikut ini merupakan rincian proses

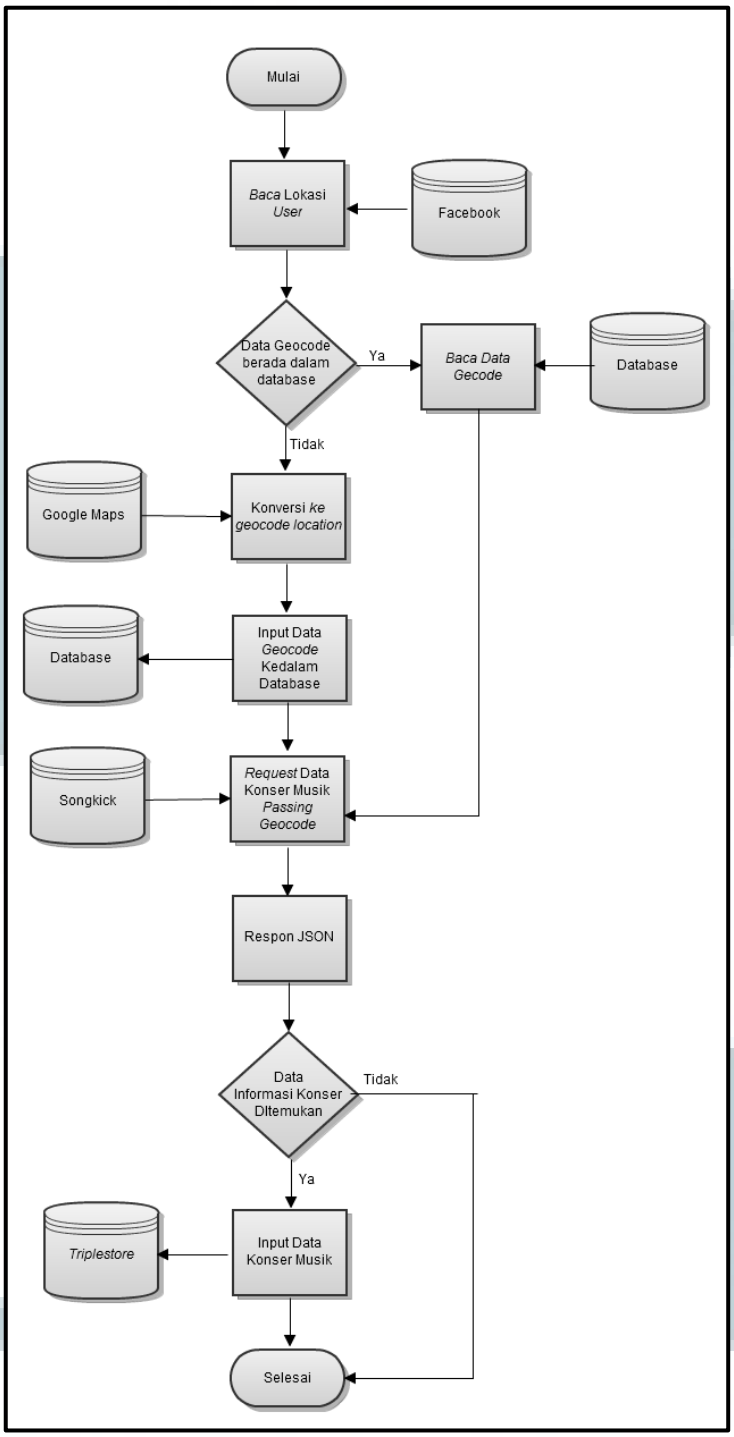
request data konser musik dari *similar artist* melalui API Songkick jika data *similar artist* ditemukan.



Gambar 3.6. Diagram *Request* Data Konser Musik *Similar Artist*

Gambar 3.6. menggambarkan proses *request* data konser musik dari *similar artist* yang diperoleh dari *knowledge base* DBpedia. Jika data konser musik ditemukan, maka data konser musik terebut akan diinput ke dalam *triplestore* melalui SPARQL *query*.

3. Diagram Alir Data *Request* Konser Musik Melalui Songkick API Dengan Parameter Lokasi

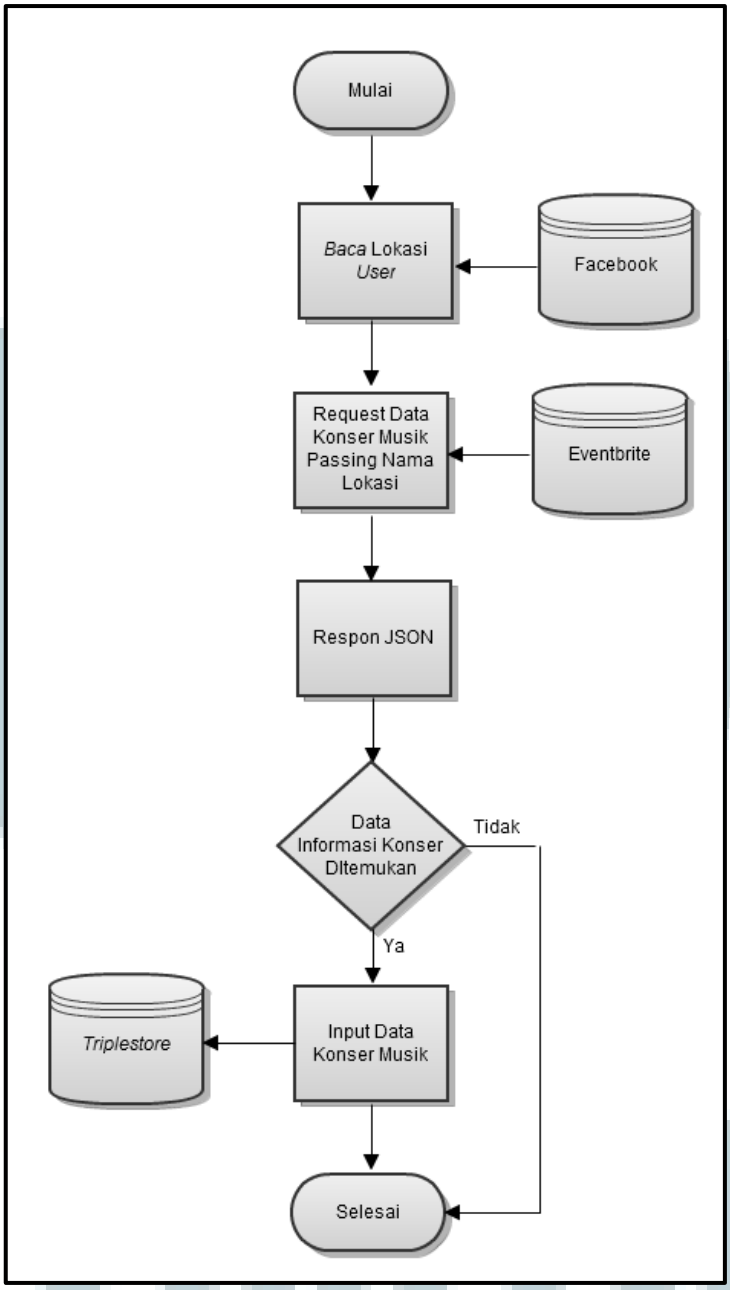


Gambar 3.7 Diagram Alir *Request* Data Konser Musik Menggunakan Parameter Lokasi dengan Format *Geocode*

Songkick hanya menerima akses parameter lokasi dalam bentuk *geocode*, sehingga dibutuhkan Google Maps API untuk konversi lokasi menjadi *geocode*. Untuk mencegah *over quota* ketika dilakukan konversi nama lokasi menjadi *geocode*, maka setiap data hasil konversi disimpan dalam *database*. Hasil konversi tersebut akan di-*passing* untuk memperoleh data konser musik berdasarkan lokasi konser.

UMN

4. Diagram Alir *Request* Konser Musik Evenbrite



Gambar 3.8. Diagram Alir *Request* Data Konser Musik pada Eventbrite Menggunakan Parameter Nama Lokasi

Untuk memperoleh data konser musik melalui API Eventbrite, diperlukan HTTP *request* dengan mem-*passing* nama lokasi. Jika data konser musik ditemukan maka data tersebut akan diinput ke dalam *triplestore* dengan SPARQL *query*.

3.3. Rancangan Ontologi

Ontologi yang digunakan untuk penelitian ini adalah ontologi Friend of a Friend (FOAF) dan Ontologi Musik. FOAF maupun Ontologi Musik memiliki *class* dan *properties*. Tahap awal perancangan ontologi adalah menyusun struktur ontologi tersebut sebagai kerangka informasi dari seluruh *vocabulary* yang telah didefinisikan oleh Ontologi FOAF dan Ontologi Musik. Untuk implementasi penelitian sistem rekomendasi konser musik ada beberapa penambahan *property* untuk menunjang kebutuhan sistem rekomendasi ini. Tidak semua *class* dan *properties* yang disediakan oleh Ontologi FOAF dan Ontologi Musik digunakan mengacu pada relevansi kebutuhan sistem.

Perancangan ontologi menggunakan *tool* Protégé 3.4.8 dengan mengimplementasikan *vocabulary* Ontologi Friend of A Friend (FOAF) dan Ontologi Musik seusai dengan kebutuhan sistem rekomendasi konser musik. Hasil rancangan tersebut diekstrak menjadi sebuah file OWL yang di-*hosting* pada server idHostinger. Untuk dapat menampilkan model ontograf atau skema hasil rancangan digunakan tool Protégé 4.3 dengan meng-*install* aplikasi *add-on* Owlviz *Plug-in* dan OntoGraf *Plug-in*. Seluruh *add-on* bersifat *open-source* dan dapat diunduh pada *website* resmi Protégé. Berikut ini merupakan penjelasan

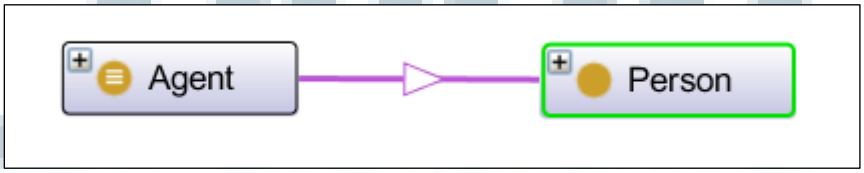
untuk merancang OWL yang sesuai dengan kebutuhan sistem rekomendasi konser musik:

1. Membuka aplikasi Protégé 3.4.8 yang telah ter-*install* pada komputer, kemudian setelah jendela terbuka pilih menu create new OWL ontology.
2. Untuk mengimplementasikan Ontologi FOAF dan Ontologi Musik, dilakukan *import specification* Ontologi FOAF yang dapat diakses di http://xmlns.com/foaf/spec/index.rdf dan *import specification* Ontologi Musik yang dapat diakses di http://raimond.me.uk/resources/mo.rdf.

Secara garis besar, *class* yang digunakan untuk memenuhi kebutuhan sistem rekomendasi konser musik yaitu *class* foaf:Person sebagai interpretasi pengguna Facebook dan *class* mo:Performance sebagai interpretasi dari *performance event* yang akan diadakan. Berikut ini merupakan penjelasan secara lebih detail class foaf:Person dan class mo:Performance.

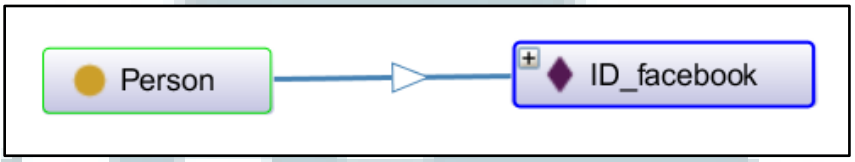
1. *Class* foaf:Person

Class yang merepresentasikan suatu agen (*person*, *group*, *software*) yang merupakan *subclass* dari class foaf:Agent.



Gambar 3.9. Relasi *Class* foaf:Agent dan *Class* foaf:Person

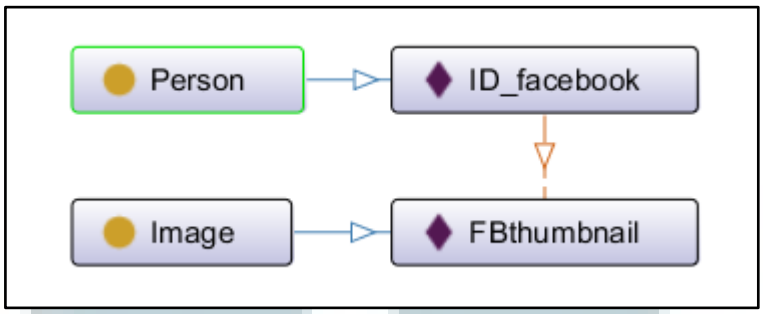
Seperti digambarkan pada Gambar 3.9 *class* foaf:Person sebagai *subclass* dari *class* foaf:Agent berperan sebagai pengguna Facebook yang memiliki *value username* dari pengguna Facebook (misal alexander.arda).



Gambar 3.10. Relasi *Class* foaf:Person dan Individu ID_Facebook

Seperti terlihat pada Gambar 3.10 Individu ID_facebook sebagai individu dari *class* foaf:Person, didesain menggunakan beberapa *property* yang dapat dikelompokkan menjadi dua bagian yaitu *property* yang sudah didefinisikan dalam *vocabulary* FOAF dan *property* yang ditambahkan secara manual untuk memenuhi kebutuhan sistem rekomendasi konser musik. Untuk *property* kelompok pertama, *property* yang sudah didefinisikan dalam *vocabulary* FOAF sebagai berikut:

1. `foaf:firstName`. *Property* yang merepresentasikan nama depan.
2. `foaf:lastName`. *Property* yang merepresentasikan nama belakang pengguna Facebook.
3. `foaf:gender`. *Property* yang merepresentasikan gender pengguna Facebook.
4. `foaf:status`. *Property* yang merepresentasikan ekspresi dan aktivitas saat ini dari pengguna Facebook.
5. `foaf:img`. *Property* yang menghubungkan *class* foaf:Person dan *class* foaf:Image.



Gambar 3.11. Relasi Individu Dari *Class* foaf:Person dan Individu dari *Class* foaf:Image

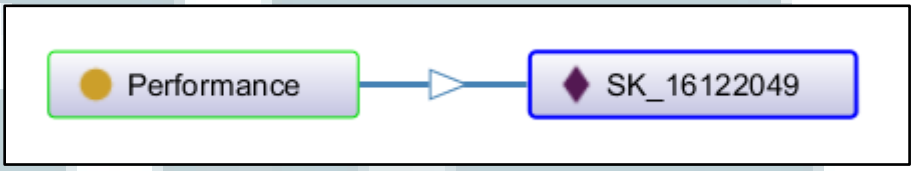
Sesuai dengan yang ditampilkan pada Gambar 3.11, melalui *property* `foaf:img` kedua individu ID_facebook dan individu Fbthumbnail akan saling terhubung.

Sedangkan untuk *property* yang ditambahan secara manual untuk memenuhi kebutuhan sistem rekomendasi konser musik adalah sebagai berikut:

1. `foaf:likedMusic`. *Property* yang ditambahkan secara manual untuk memenuhi kebutuhan sistem rekomendasi konser musik.
2. `foaf:hometown`. *Property* yang ditambahkan untuk merepresentasikan kampung halaman pengguna Facebook.
3. `foaf:location`. *Property* yang merepresentasikan lokasi sekarang pengguna Facebook.
4. `foaf:country`. *Property* yang merepresentasikan negara asal pengguna Facebook.

2. *Class* mo:Performance

Class yang merepresentasikan suatu penampilan pertunjukan di mana dalam pertunjukan tersebut terdapat *performer*, *musical instrument*, dan menghasilkan suara musikal. Pada implementasi penelitian ini, *class* mo:Performance memiliki individu yang akan dinamakan sesuai dengan *identifier* yang diperoleh dari API penyedia informasi konser musik (misal SK28909504).

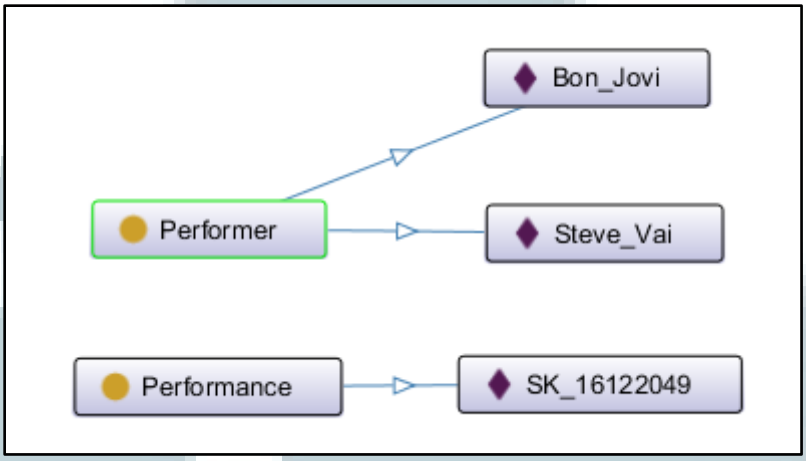


Gambar 3.12. Relasi Individu Dari *Class* foaf:Person dan Individu dari *Class* foaf:Image

Gambar 3.12, untuk memenuhi kebutuhan sistem, diperlukan beberapa *custom property* untuk melengkapi *property* Ontologi Musik yang telah disediakan. Berikut ini *property* yang digunakan dan sudah didefinisikan pada *vocabulary* Ontologi Musik sebagai berikut.

1. `mo:event_homepage`. *Property* yang merepresentasikan alamat URL suatu *event* yang diselenggarakan. Untuk memungkinkan penyimpanan data dalam bentuk URI, maka digunakan enkripsi sha1.
2. `mo:genre`. *Property* yang digunakan untuk menghubungkan *class* mo:performance dengan *class* `mo:Genre` melalui individu *performance* dan individu dari *class* Genre.

3. `mo:performer`. *Property* yang digunakan untuk menghubungkan individu dari *class* mo:Performance dengan individu dari *class* mo:Performer.



Gambar 3.13. Relasi *Class* mo:Performer dengan Individu dan *Class* mo:performance dengan Individu

Gambar 3.13 menunjukkan relasi antar *class* mo:Performer terhadap individu dari *class* mo:Performer dan relasi *class* mo:Performance dengan individu dari *class* mo:Performance. Pada Gambar 3.11 mengambarkan suatu pertunjukkan dengan *identifier* SK_16122049 memiliki *performer* dengan *identifier* Bon_Jovi dan Steve_Vai, untuk menghubungkan kedua individu tersebut digunakan *property* mo:performer.

Sedangkan di bawah ini, *custom property* yang ditambahkan untuk memenuhi kebutuhan sistem rekomendasi musik, yaitu:

1. `mo:performanceName`. *Property* yang digunakan untuk merepresentasikan nama suatu *performance*.
2. `mo:city`. *Property* yang digunakan untuk merepresentasikan kota lokasi penyelenggaraan *performance*.

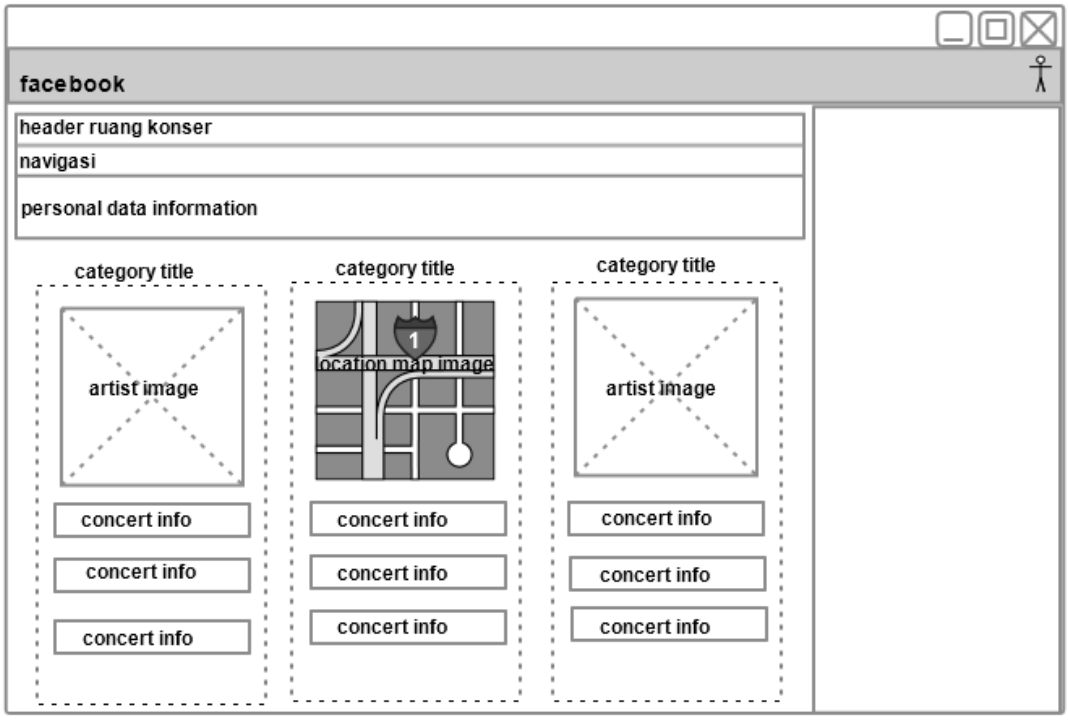
3. `mo:description`. *Property* yang merepresentasikan deskripsi suatu *performance*.
4. `mo:startTime`. *Property* yang merepresentasikan waktu dimulainya suatu *performance*.
5. `mo:price`. *Property* yang merepresentasikan harga tiket suatu *performance*.
6. `mo:venue`. *Property* yang digunakan untuk merepresentasikan venue suatu *performance*.
7. `mo:vendor`. *Property* yang digunakan untuk merepresentasikan vendor dari penyedia informasi konser.
8. `mo:date`. *Property* yang merepresentasikan tanggal suatu *performance*.
9. `mo:country`. *Property* yang merepresentasikan negara penyelenggaraan *performance*.
10. `mo:performanceID`. *Property* yang digunakan untuk merepresentasikan *identifier* dari suatu *performance*.

3.4. Perancangan Antarmuka

Rancangan tampilan antarmuka aplikasi dibuat untuk aplikasi yang akan ditampilkan pada kanvas Facebook menggunakan iFrame. Untuk tampilan *popup login* menyesuaikan tampilan dari Facebook.

Tampilan antarmuka pada aplikasi sistem rekomendasi konser musik ditampilkan dalam satu halaman memanjang vertikal yang terbagi menjadi lima *section*. Dengan demikian tampilan antarmuka dirinci sebagai berikut:

1. ***Section Home***



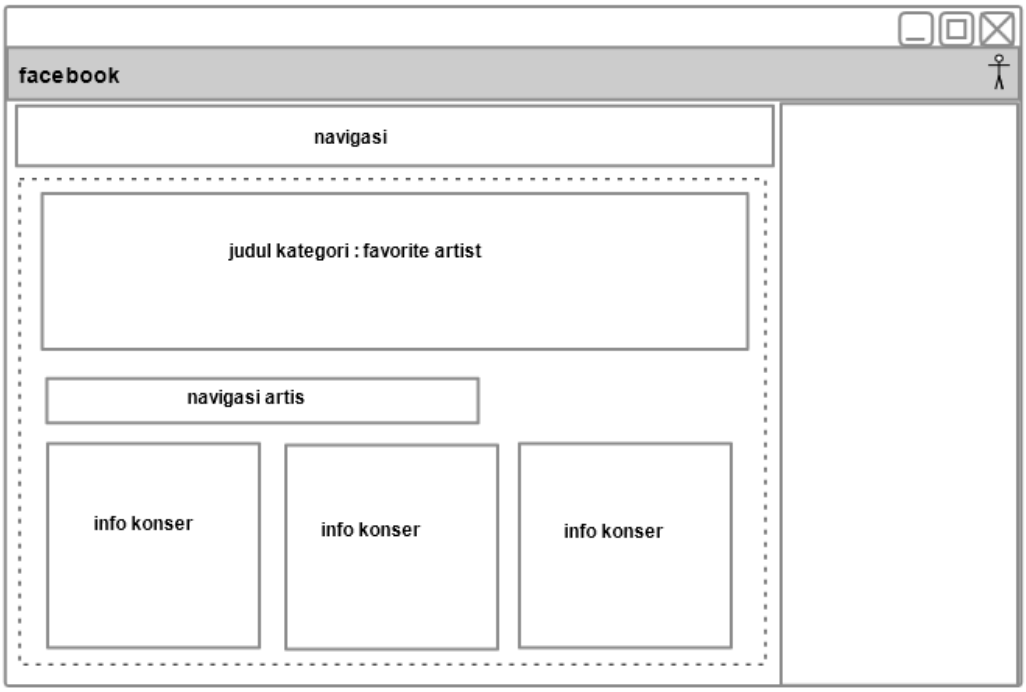
Gambar 3.14. Rancangan Tampilan Antarmuka *Section Home*

Gambar 3.14 menjelaskan tampilan antar muka pada *section home*. Seluruh halaman aplikasi yang ditampilkan pada aplikasi Facebook akan berada di dalam sebuah border milik Facebook. *Section* home menjadi halaman pertama yang akan ditampilkan kepada *user*. Menampilkan navigasi, personal data pengguna aplikasi, dan akan menampilkan daftar konser sesuai dengan kategori masing-masing: *Your Favorite*, *Near You*, dan *Suggested Concert*. Untuk kategori

Your Favorite dan *Suggested Concert* akan ditampilkan foto artis secara acak yang diambil dari Facebook Graph API. Sedangkan untuk kategori *Near You* akan ditampilkan gambar tampak daerah yang diambil dari Google Maps API.

Daftar konser yang ditampilkan pada *section Home* tidak menampilkan seluruh data daftar konser musik. Daftar konser musik selengkapnya akan ditampilkan per-*section*.

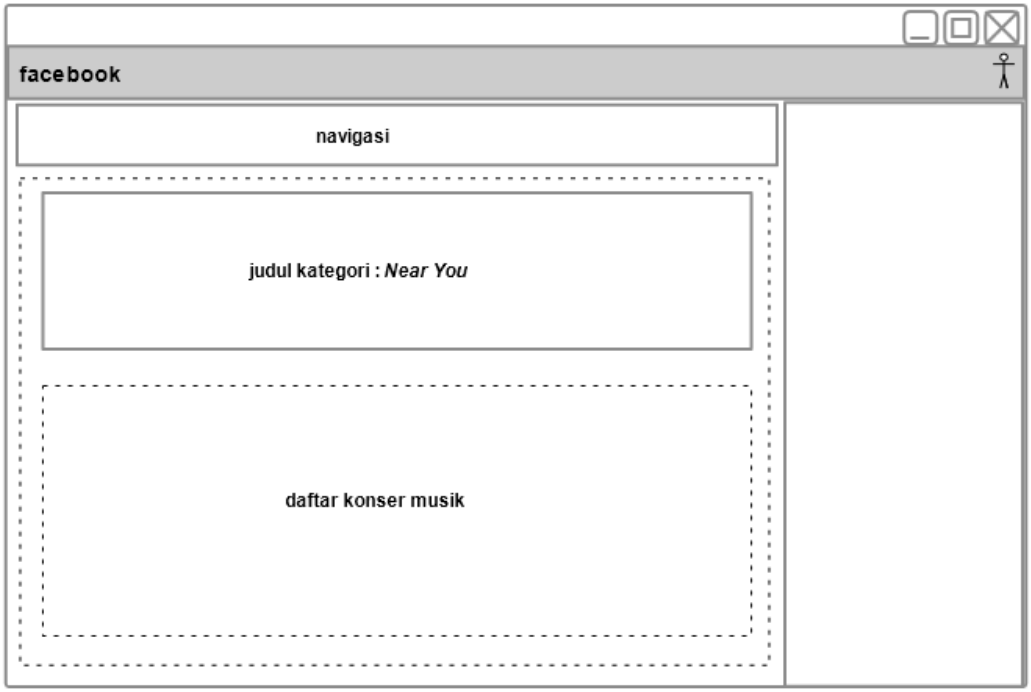
2. *Section Your Favorite*



Gambar 3.15. Rancangan Tampilan Antarmuka *Section Your Favorite*

Pada *section favorite artist* seperti digambarkan oleh Gambar 3.15, ditampilkan *button* navigasi nama artis yang jika diklik akan menampilkan seluruh informasi konser musik artis tersebut.

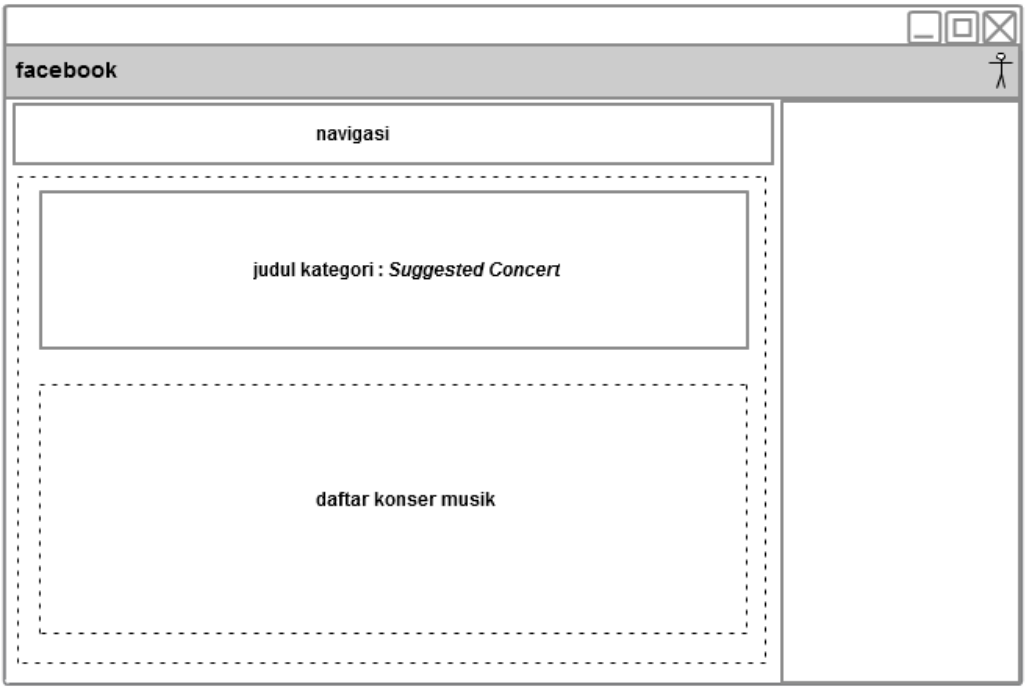
3. *Section Near You*



Gambar 3.16. Rancangan Tampilan Antarmuka *Section Near You*

Gambar 3.16 menggambarkan rancangan tampilan antar muka untuk section *Near You*. Pada *section Near You*, akan menampilkan daftar seluruh konser musik berdasarkan lokasi *user*.

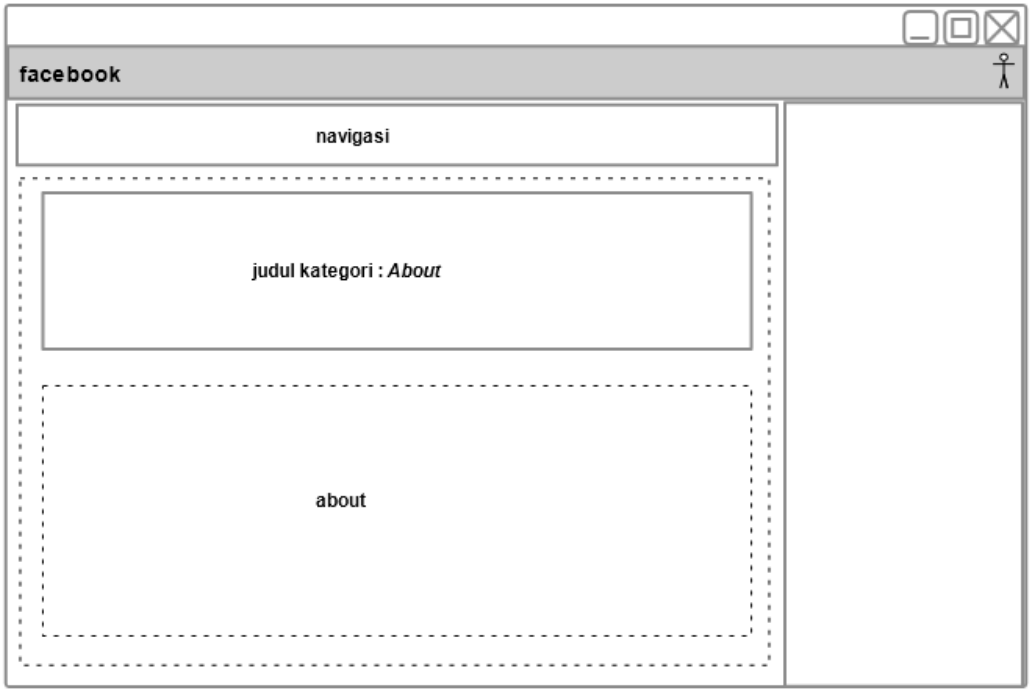
4. *Section Suggested Concert*



Gambar 3.17. Rancangan Tampilan Antarmuka *Section Suggested Concert*

Gambar 3.17 menggambarkan rancangan tampilan antar muka pada *section Suggested Concert*. Pada *section* ini akan ditampilkan seluruh daftar konser berdasarkan artis hasil rekomendasi sistem rekomendasi musik.

5. *Section About*



Gambar 3.18. Rancangan Tampilan Antarmuka *Section About*

Rancangan tampilan antar muka *section About*, seperti ditampilkan pada Gambar 3.18 akan menampilkan profil sederhana mengenai aplikasi sistem rekomendasi konser musik.

3.2.3. Metode Pengujian dan Penilaian

Untuk mencapai tujuan penelitian, dilakukan pengujian fungsionalitas umum dan penilaian kegunaan dari aplikasi sistem rekomendasi konser musik pada *platform* Facebook.

3.2.3.1. Pengujian Fungsionalitas Umum

Tujuan pengujian ini adalah untuk mengetahui apakah fitur-fitur aplikasi telah berjalan sesuai yang diharapkan dan adanya kesesuaian antara data pada Facebook dan *output* rekomendasi data konser musik. Pengujian dilakukan dengan menjalankan tiap fitur yang terdapat pada sistem dan mencatat hasil yang dikeluarkan oleh tiap fitur tersebut. Setiap fungsi pada aplikasi sistem rekomendasi ini akan diuji sebanyak sepuluh kali dengan kondisi acak.

3.2.3.1. Pengujian Skala Kegunaan Sistem

Pengujian kesesuaian *output* dilakukan dengan menerapkan metode System Usability Scale (SUS). SUS merupakan sebuah metode untuk mengetahui kegunaan suatu aplikasi berbasis web. SUS terdiri dari sepuluh pernyataan, masing-masing memiliki skala lima poin yang memiliki rentang dimulai dari sangat tidak setuju hingga sangat setuju. Dibagi menjadi lima pernyataan positif dan lima pernyataan negatif untuk duapuluh responden yang dipilih secara acak.

Perhitungan *score* hasil kuesioner dilakukan dengan cara sebagai berikut:

1. Untuk pernyataan nomor ganjil, kurangkan hasil respon dengan angka 1.
2. Untuk pernyataan nomor genap, jumlah didapatkan dengan mengurangkan angka 5 dengan hasil respon.
3. Skala hasil perhitungan pernyatan memiliki rentang 0 sampai dengan 4, dengan nilai 4 merupakan nilai tertinggi.
4. Jumlahkan seluruh hasil kemudian kalikan dengan 2,5. Hasil akhir akan memiliki skala 0 sampai dengan 100.

Pada penelitian ini, sepuluh pernyataan pada SUS telah disesuaikan dengan kebutuhan penilaian untuk aplikasi sistem rekomendasi konser musik tanpa menghilangkan esensi utama dari SUS.

KUESIONER
Sistem Rekomendasi Konser Musik
Ruang Konser

Nama :
Usia :

Silahkan beri tanda cek pada kotak yang menggambarkan penilaian Anda untuk setiap pernyataan di bawah ini.

		Sangat Tidak Setuju				Sangat Setuju
No	**Pernyataan**	**1**	**2**	**3**	**4**	**5**
1.	Saya merasa sangat nyaman menggunakan aplikasi ini.					
2.	Saya menemukan kerumitan ketika menggunakan sistem, yang sebenarnya tidak diperlukan.					
3.	Menurut saya aplikasi ini mudah digunakan.					
4.	Saya membutuhkan bantuan dari orang teknis untuk menggunakan aplikasi ini.					
5.	Saya menemukan berbagai fungsi dalam aplikasi yang terintegrasi dengan baik.					
6.	Saya pikir ada banyak ketidakkonsistenan dalam aplikasi ini.					
7.	Saya berpikir sebagian besar orang akan bisa menggunakan aplikasi ini dengan cepat.					
8.	Saya menemukan kejanggalan dalam penggunaan aplikasi ini.					
9.	Saya akan sering menggunakan aplikasi ini untuk mencari informasi konser musik.					
10.	Saya perlu belajar banyak hal sebelum saya terbiasa dengan aplikasi ini.					

Gambar 3.19. Format Kuesioner

Gambar 3.19 menampilkan kuesioner untuk penilaian kegunaan aplikasi sistem rekomendasi konser musik dengan mengimplementasikan Ontologi *Friend of A Friend* dan Ontologi Musik.