

Rekomendasi Rute Kunjungan Tempat Wisata Menggunakan Ontologi dan Algoritma A*

¹Nurul Fajrin Ariyani, ²Ivan Fadhila, ³Abdul Munif

Departemen Teknik Informatika, Institut Teknologi Sepuluh Nopember,
Surabaya, Indonesia

Email: ¹nurulfajrin@if.its.ac.id, ²ivan.fadhila70@gmail.com, ³munif@if.its.ac.id

Tersedia Online di

<http://www.jurnal.unublitar.ac.id/index.php/briliant>

Sejarah Artikel

Diterima pada 17 April 2020
Disetujui pada 24 April 2020
Dipublikasikan pada 30 Mei 2020
Hal. 417-430

Kata Kunci:

Algoritma A*; ontology; rute perjalanan wisata; sistem rekomendasi; SPARQL

DOI:

<http://dx.doi.org/10.28926/briliant.v3i4.459>

Abstrak: Meningkatnya sektor pariwisata di Indonesia menjadikan para wisatawan memiliki banyak pilihan tempat wisata untuk dikunjungi. Namun keterbatasan waktu seringkali menghalangi wisatawan untuk dapat mengunjungi beberapa tempat wisata yang berbeda dalam satu kali rute perjalanan. Sistem rekomendasi yang dibangun memanfaatkan *OpenStreetMap* dan *PostgreSQL* sebagai basis data spasial untuk menyimpan data spasial mengenai peta. Jarak terdekat antar lokasi dicari dengan menerapkan algoritma A*. Sedangkan ontologi dan SPARQL digunakan untuk menyimpan dan mendapatkan kembali tempat-tempat wisata sesuai dengan kategori dan sub-kategorinya. Sistem diujikan dengan menggunakan studi kasus tempat-tempat wisata di kota Surabaya. Dari hasil uji coba dapat disimpulkan bahwa sistem mampu menampilkan rekomendasi rute kunjungan wisata dari suatu titik lokasi awal ke lokasi tujuan, beserta tempat-tempat wisata yang dilalui oleh rute perjalanan tersebut sesuai dengan kategori wisata yang dipilih oleh pengguna.

PENDAHULUAN

Meningkatnya sektor pariwisata di Indonesia menjadikan para wisatawan memiliki banyak pilihan tempat wisata untuk dikunjungi. Namun, keberagaman tempat wisata ini tidak jarang membuat wisatawan mengalami kesulitan untuk menentukan lokasi wisata yang ingin dikunjungi dikarenakan keterbatasan waktu yang dimiliki. Aplikasi penunjuk rute perjalanan seperti Google Maps bisa dijadikan wisatawan sebagai pertimbangan untuk memilih rute mana yang ingin mereka lalui berdasarkan moda transportasi yang tersedia. Namun aplikasi tersebut tidak dapat memberikan rekomendasi rute apabila wisatawan memiliki preferensi lain untuk mengunjungi tempat wisata tertentu di sepanjang rute perjalanan yang akan mereka lalui.

Pada umumnya wisatawan telah memiliki preferensi kategori tempat wisata tertentu yang ingin dikunjungi berdasarkan pengalaman dari perjalanan sebelumnya (Sebastia *et al.*, 2009). Namun kurangnya pengetahuan tentang kota yang dikunjungi seringkali menyulitkan wisatawan untuk mengatur rute kunjungan secara efisien. Beberapa penelitian sebelumnya telah menghasilkan aplikasi rekomendasi wisata yang beragam tergantung dari kebutuhan masing-masing pengguna. Sistem yang dibuat oleh (Pazzani, 1999) mempertimbangkan

lokasi wisatawan dan minat kategori wisata untuk menghasilkan rekomendasi rute. Keuntungan dari sistem ini adalah selalu dapat memberikan rekomendasi, namun hasilnya kurang akurat karena tidak disertai informasi pendukung mengenai tempat wisata tersebut. Sebuah rekomendasi wisata seharusnya tidak hanya mempertimbangkan preferensi wisatawan saja, tetapi juga menyajikan informasi tentang keadaan tempat tersebut seperti jarak tempuh, moda transportasi, jam buka, dan sebagainya (Delgado and Davidson, 2002). Penerapan ontologi untuk melakukan penalaran dalam menghasilkan usulan tempat wisata dengan mempertimbangkan lokasi, kemacetan, pemandangan dan motivasi kunjungan dari wisatawan terbukti meningkatkan keakuratan dari rekomendasi (Daramola, Adigun and Ayo, 2009; Moreno *et al.*, 2013). Namun secara teknis, proses penalaran tersebut membutuhkan sumber daya mesin yang tidak sedikit serta memakan waktu proses yang lama (Batsakis and Petrakis, 2010).

Berdasarkan pada permasalahan tersebut, peneliti membangun sebuah sistem rekomendasi berbasis web yang dapat memberikan rekomendasi rute perjalanan dan kunjungan wisata sesuai dengan kategori tempat wisata yang dipilih oleh pengguna. Data mengenai tempat wisata disimpan ke dalam basis data dan juga penyimpanan lain berskema ontologi. Implementasi ontologi bertujuan untuk mengenali makna taksonomi dari kategori wisata dan subkategori-subkategori yang ada di bawahnya. Sering kali satu tempat wisata yang identik bisa dikelompokkan ke dalam dua atau lebih kategori wisata yang berbeda. Fakta semacam ini secara tertulis dapat disimpan secara sederhana menggunakan basis data relasional, namun makna dari kategori tersebut tidak dapat disimpan secara utuh karena basis data membatasi relasi antar kategori wisata satu dengan lainnya secara kaku (Astrova, Korda and Kalja, 2007).

Selain dapat memberikan rekomendasi rute kunjungan, sistem juga dapat memberikan informasi terkait dengan lingkungan tempat wisata seperti jam buka maupun perkiraan lamanya waktu kunjungan. Peneliti menerapkan teknologi *linked open data* dengan cara menghubungkan setiap entitas tempat wisata yang ada di basis data dengan sumber informasi metadata yang ada pada Dbpedia (Lehmann *et al.*, 2015). Dbpedia menyediakan informasi penunjang lain mengenai tempat wisata di luar jam buka, seperti sejarah, kondisi, hingga ciri khas dari tempat wisata tersebut seperti yang tercantum pada laman Wikipedia.

Untuk membangun visualisasi rute perjalanan, peneliti menggunakan data peta yang disediakan oleh OpenStreetMap (OSM), yaitu penyedia layanan peta berbasis *open source*. Peta tersebut akan dikonversi ke dalam basis data spasial menggunakan PostgreSQL. Sedangkan untuk mencari rute kunjungan secara efisien, peneliti mengimplementasikan algoritma A* berdasarkan pertimbangan bahwa algoritma tersebut dapat diaplikasikan pada area topologis seperti pada peta jalan (Cui, Wang and Yang, 2012). Algoritma A* menggabungkan antara pencarian heuristik dan pencarian jarak terdekat untuk memastikan bahwa rute yang terpilih merupakan rute dengan jarak tempuh minimal yang melewati titik-titik lokasi tempat wisata sesuai pilihan wisatawan.

Sistem rekomendasi ini bertujuan membantu wisatawan atau pengguna untuk mendapatkan arahan rute perjalanan dari suatu titik awal ke lokasi akhir yang melewati tempat-tempat wisata lain yang diminatinya. Sistem dibangun berbasis web untuk memudahkan pengoperasian oleh pengguna umum. Dalam jangka waktu panjang, sistem diharapkan dapat memberikan kontribusi bagi dinas

pariwisata lokal terkait untuk menarik minat dan memberikan kemudahan pada pengunjung wisata.

METODE

Bab ini menjabarkan metode yang digunakan untuk menjawab permasalahan pada sistem rekomendasi rute perjalanan. Secara garis besar, penjelasan metode akan dibagi menjadi tiga subbab, yaitu: persiapan dataset, pembangunan ontologi, dan deskripsi umum sistem. Masing-masing subbab akan dijabarkan sebagai berikut.

A. Persiapan Dataset

Dataset yang digunakan sistem terbagi menjadi dua jenis, yaitu data peta dan data tempat wisata. Data peta didapatkan dari *OpenStreetMap (OSM)*, yaitu sebuah proyek kolaboratif yang bertujuan untuk mengumpulkan geodata tentang dunia. OSM memiliki komunitas kontributor yang memperbaharui peta secara berkala. Peta atau geodata tersebut tersedia secara gratis dan dapat diunduh tanpa dipungut biaya. Geodata OSM memiliki tiga tipe data, antara lain *nodes*, *ways*, dan *relations*. *Nodes* merupakan titik yang berisi koordinat, versi, serta tanggal titik tersebut dibuat. *Ways* berupa garis yang merepresentasikan jalan, terdiri dari tiga jenis, yaitu *open way*, *closed way*, dan *area* yang berbentuk poligon. Sedangkan *relations* merupakan hubungan antar elemen pada data OSM.

Peneliti memasukkan geodata OSM ke dalam basis data spasial PostgreSQL dengan bantuan *Osm2pgrouting* agar dapat diproses oleh sistem rekomendasi. *Osm2pgrouting* merupakan sebuah perangkat yang digunakan untuk memasukkan data peta dari OSM ke dalam basis data spasial PostgreSQL (*osm2pgrouting - Import OSM data into pgRouting Database — Open Source Routing Library*, no date). Perangkat ini digunakan untuk memilih tipe jalan dan kelas yang akan dimasukkan ke basis data spasial. Selain itu, *osm2pgrouting* juga membangun tabel dan tipe datanya secara otomatis pada basis data sehingga sangat memudahkan proses perhitungan jarak (*cost*) pada aplikasi.

Jenis dataset yang kedua adalah dataset tempat wisata. Dataset wisata pada penelitian ini merupakan gabungan data yang didapatkan dari beberapa alamat situs, seperti Pemerintah Kota Surabaya¹, East Java², Surabaya Tourism³, serta beberapa sumber lainnya. Properti data yang diambil berupa nama tempat wisata, *latitude*, *longitude*, alamat, jam buka, jam tutup, serta kategori tempat wisata tersebut. Data yang ada kemudian dimasukkan ke dalam satu tabel dengan nama *places*. Untuk properti kategori tempat wisata nantinya akan disimpan menggunakan skema ontologi yang dijabarkan pada tahap selanjutnya. Tabel 1 menunjukkan contoh properti data tempat wisata beserta nilainya. Kolom *Tempat* memuat nama tempat wisata, *latitude* dan *longitude* merepresentasikan titik koordinat yaitu garis lintang dan garis bujur, sedangkan *kategori* berisi kategori wisata dari tempat tersebut.

¹ <https://surabaya.go.id/>

² <https://www.eastjava.com/>

³ <http://www.surabayatourism.com/>

Tabel 1. Contoh Nilai dari Properti Tempat Wisata

Tempat	Latitude (x)	Longitude (y)	Kategori
Galaxy Mall Surabaya	-7.27453	112.78203	Mal
Gereja Katolik Santa Perawan Maria	-7.242042	112.7369953	Gereja
House of Sampoerna	-7.2309923	112.7341461	Museum
Hutan Bambu Keputih	-7.29854	112.80437	Hutan

B. Pembangunan Ontologi

Skema ontologi ditujukan untuk memaknai kategori wisata berdasarkan hierarki *class* beserta *subclass*-nya. Berbeda dengan skema basis data relasional yang cenderung kaku dan statis, skema ontologi ini sangat fleksibel dan memungkinkan adanya modifikasi skema di kemudian hari meskipun skema tersebut telah memiliki baris data (*instances*).

Ontologi merupakan salah satu dasar terpenting dalam *semantic web* (Euzenat, Roşoiu and Trojahn, 2013). Ontologi merupakan disiplin ilmu yang membahas tentang sistek taksonomi metadata. Ontologi mengatur cara penamaan, tipe, property, dan hubungan antar entitas pada domainnya masing-masing. Di dalam ontologi terdapat beberapa komponen seperti:

(a) *Class*

Class adalah kelompok dari *instance*. *Class* mencakup *superclass* dan *subclass* yang juga dapat diilustrasikan menjadi hierarki kelas atau taksonomi. *Class* mewakili kategori tempat wisata, misalnya mal, museum, hutan, tempat ibadah, dan lain sebagainya.

(b) *Instance*

Instance adalah suatu individua tau objek yang juga merupakan anggota dari sebuah *class*. Sebuah *instance* menggambarkan suatu tempat wisata yang spesifik, seperti Galaxy Mall Surabaya, House of Sampoerna, Hutan Bambu Keputih, dan lain-lain.

(c) *Property*

Property menjelaskan tentang karaktersistik dari *instance* yang menjadi anggota dari sebuah *class*. Selain itu, *property* juga menggambarkan hubungan antar *instance*. Skema ontologi dalam penelitian ini hanya menggunakan satu jenis *object property* yaitu `rdfs:subClassOf`.

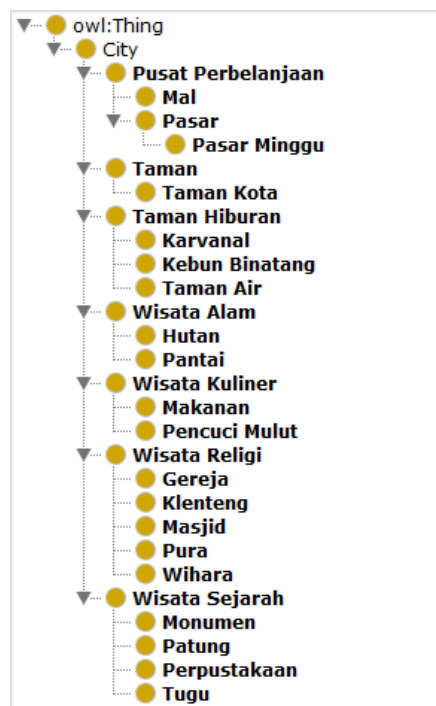
Ontologi pada penelitian ini dibangun dengan menggunakan perangkat pembantu yang dinamakan Protégé. Berdasarkan pada data tempat wisata yang berhasil dikumpulkan, peneliti mengkategorikan tempat wisata ke dalam 7 kategori utama (*class*) dan 13 subkategori (*subclass*) di bawahnya. Peneliti menambahkan *label* di setiap kategori dan subkategori tersebut agar skema ontologi dapat dihubungkan dengan basis data PostgreSQL berdasarkan terminologi yang telah disepakati keduanya. Taksonomi kategori wisata yang dibangun ditunjukkan pada Gambar 1. Tiap titik pada Gambar 1 merepresentasikan *class* atau nama kategori wisata, sedangkan titik yang berada pada tingkat dibawahnya menandakan *subclass*. Sebagai contoh, kategori wisata Pusat Perbelanjaan memiliki subkategori Mal dan Pasar, sedangkan kategori Pasar memiliki subkategori Pasar Minggu. Dengan demikian, apabila kueri meminta untuk menampilkan tempat wisata berupa Pusat Perbelanjaan, maka semua

instance yang ada pada kategori Pusat Perbelanjaan beserta semua subkategorinya akan ditampilkan sebagai hasil.

Skema ontologi beserta *instance* datanya disimpan dalam format *Resource Description Framework* (RDF). RDF menggunakan *Universal Resource Identifier* (URI) untuk mengidentifikasi sebuah *resource* atau benda. RDF adalah kerangka kerja untuk menggambarkan sumber daya web. RDF menjadi metode umum untuk mendeskripsikan sumber daya web sehingga dapat dibaca dan dipahami oleh aplikasi komputer. Akar mula atau *root* dari sebuah ontologi disebut dengan *Thing*, seperti ditunjukkan pada Gambar 1.

RDF memiliki tiga elemen utama, yaitu subjek, objek, dan predikat. Sebuah <subjek> dapat memiliki properti <predikat> yang menjelaskan hubungan subjek tersebut dengan sebuah <objek>. Nilai dari ketiga elemen tersebut berupa URI atau link sumber daya web. Gambar 2 merupakan contoh sintaks RDF dari “Galaxy Mall Surabaya” seperti ditunjukkan pada Tabel 1. Dengan rincian sebagai berikut:

- 1) <subjek>, berupa URI sumber daya web yang menjelaskan tentang tempat wisata bernama Galaxy Mall Surabaya. *Resource* tempat wisata mengambil dari domain Dbpedia Indonesia, yang merupakan sumber metadata artikel yang ada pada Wikipedia. Penulis menggunakan sumber dari Dbpedia dikarenakan memiliki properti metadata yang cukup lengkap, selain properti koordinat dan nama tempat. Properti ini kemungkinan akan dibutuhkan di penelitian selanjutnya.
- 2) <predikat>, berupa properti yang menjelaskan letak koordinat dari <subject>. Predikat dalam skema ontologi ini terdiri dari *hasLat* dan *hasLong* yang menyatakan nilai latitude dan longitude dari <subject>.



Gambar 1. Taksonomi Kategori Wisata

- 3) <objek>, seperti “-7.27453” dan “112.78203” yang menyatakan sebuah nilai, dalam hal ini berupa string.

Selain resource yang disebutkan dalam `rdf:about`, sintaks berikut juga memiliki label yang berisi nama resmi dari tempat tersebut, dalam bentuk string. Dengan demikian, dapat dijelaskan bahwa resource menceritakan link URI dari sumber daya web yang dirujuk, sedangkan label merepresentasikan nama resmi dari objek yang dibahas pada resource tersebut. *Label* inilah yang akan dicocokkan dengan kolom nama tempat yang ada pada basis data PostgreSQL.

C. Deskripsi Umum Sistem

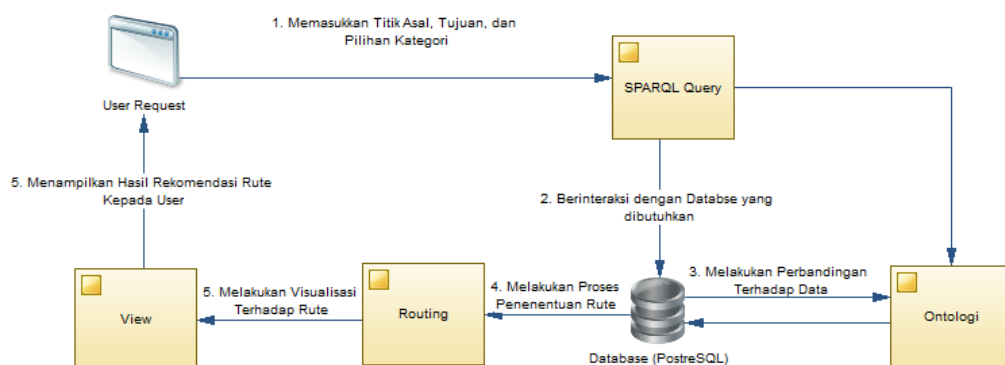
Sistem rekomendasi memiliki arsitektur yang terbagi menjadi enam tahapan, yaitu: (1) memasukkan titik awal, titik tujuan, dan pilihan kategori; (2) melakukan perbandingan terhadap data; (3) melakukan kueri pencarian; (4) melakukan proses penentuan rute; (5) melakukan visualisasi terhadap rute; dan (6) menampilkan hasil rekomendasi rute kepada pengguna. Adapun bentuk diagram dari arsitektur sistem ditunjukkan pada Gambar 3. Kotak kuning menunjukkan komponen yang menangani masing-masing tahapan proses. Sedangkan tabung berwarna abu-abu menandakan database yang digunakan, yaitu PostgreSQL. Gambar 3 juga menceritakan jalannya alur tahapan proses yang berjalan pada sistem mulai dari pengguna memasukkan data awal, hingga visualisasi akhir rute rekomendasi kunjungan. Penjabaran dari masing-masing tahapan akan dijelaskan sebagai berikut:

- 1) Memasukkan titik awal, titik tujuan, dan pilihan kategori.

Pada tahap pertama, pengguna sistem diminta untuk memasukkan titik awal perjalanan, titik tujuan, dan pilihan kategori wisata yang ingin dikunjungi dalam perjalanan tersebut. Titik awal dan titik tujuan yang dimasukkan oleh

```
<owl:NamedIndividual
rdf:about="http://id.dbpedia.org/resource/rekomwisata#Galaxy_Mall_Surabaya">
  <rdf:type rdf:resource="http://www.semanticweb.org/rekomwisata#Mal"/>
  <hasLat
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">-7.27453</hasLat>
  <hasLong
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">112.78203</hasLong>
  <rdfs:label>Galaxy Mall Surabaya</rdfs:label>
</owl:NamedIndividual>
```

Gambar 2. Contoh Sintaks RDF untuk "Galaxy Mall Surabaya".



Gambar 3. Arsitektur Sistem Rekomendasi Rute Kunjungan Wisata

pengguna adalah dalam bentuk nama tempat (bukan koordinat).

2) Berinteraksi dengan basis data yang dibutuhkan.

Sistem akan mencari nama tempat tersebut di dalam database PostgreSQL untuk mendapatkan titik *latitude* dan *longitued* dari titik-titik tersebut. Sedangkan untuk kategori wisata, sistem telah menyediakan pilihan kategori dalam bentuk *dropdown list* guna meminimalisir kesalahan penulisan kategori oleh pengguna. Pengguna dapat memilih lebih dari satu kategori wisata yang berbeda, sesuai dengan preferensi kunjungan yang ingin dilakukannya.

3) Melakukan perbandingan terhadap data.

Selanjutnya sistem akan melakukan pencarian tempat-tempat wisata terlebih dahulu berdasarkan pilihan kategori wisata pengguna dengan menjalankan modul *query engine* yang ada di dalam komponen “SPARQL Query”. Pellet reasoner⁴ digunakan sebagai *query engine* dalam komponen tersebut. Sintaks SPARQL berbeda dengan sintaks kueri SQL pada umumnya. Perbedaannya, sintaks kueri SQL dikenakan pada sekumpulan baris data yang tersimpan dalam basis data relasional, sedangkan sintaks SPARQL dikenakan pada sekumpulan *instances* yang tersimpan dalam ontologi. Ontologi tersebut umumnya berbentuk file dengan ekstensi turunan dari XML, seperti OWL atau RDF. Dalam penelitian ini, peneliti menyimpan skema ontologi ke dalam format RDF untuk meringankan beban kerja dari *query engine*.

Untuk memanfaatkan fungsi taksonomi dari kategorisasi wisata yang telah dibuat, maka SPARQL kueri harus dapat mengembalikan nilai semua nama tempat wisata yang ada pada kategori tersebut maupun yang ada pada subkategorinya. Dengan demikian, bentuk sintaks SPARQL yang dijalankan adalah seperti pada **Error! Reference source not found.** Sintaks memanfaatkan properti `rdfs:subClassOf` untuk mendapatkan semua *class* kategori di bawah kategori kunci pencarian, dan berlaku secara rekursif. `?x0` berisi URI dari sumber daya web tentang tempat wisata, sedangkan `?x1` berisi label nama resmi dari tempat wisata tersebut. Hasil kueri dari `?x1` akan dicocokkan dengan baris data di PostgreSQL untuk mendapatkan titik koordinatnya masing-masing.

Untuk membandingkan data kategori pada tempat wisata di PostgreDQL dan data kategori pada skema ontologi, peneliti menggunakan bahasa *Python* dengan pustaka *RDFLib*. Pustaka tersebut digunakan untuk mengambil data kategori dan subkategori dari RDF ontologi, kemudian menjalankan sintaks kueri SQL pada PostgreSQL untuk mengambil data kategori dari tabel “Places”.

4) Melakukan proses penentuan rute.

Untuk menghindari sistem agar tidak memilih tempat wisata yang terlalu jauh dari titik awal dan titik tujuan, peneliti menggunakan batasan peta yang disebut dengan *bounding box*. Batasan peta diambil dari jarak terjauh *latitude* dan *longitude* berdasarkan rute awal yang telah ditentukan. Gambar 5 mengilustrasikan batasan peta yang diwakili dengan kotak berwarna merah. Ujung rute merupakan titik awal dan titik tujuan yang dimasukkan oleh pengguna ke dalam sistem. Pilihan tempat-tempat wisata yang akan diperhitungkan sebagai titik koordinat evaluasi pada algoritma A* adalah titik-titik lokasi tempat wisata yang berada di dalam *bounding box* saja.

⁴ <https://github.com/stardog-union/pellet>

Setelah titik-titik koordinat dari tempat wisata berhasil didapatkan, maka tahap selanjutnya adalah menggunakan titik-titik tersebut untuk menentukan rute kunjungan perjalanan wisata. Peneliti mengimplementasikan algoritma A* (dibaca: *A star*) untuk menentukan rekomendasi rute tersebut. Algoritma A* merupakan pengembangan dari algoritma Dijkstra yang menerapkan *best-first search* untuk mencari semua kemungkinan rute perjalanan mencapai tujuan akhir dengan jarak terpendek. Pada dasarnya, A* bekerja dengan cara mengevaluasi setiap kemungkinan rute yang bisa dihasilkan dari rangkaian sekumpulan titik koordinat yang diamati, dalam hal ini berupa titik lokasi tempat wisata sesuai dengan kategori wisata yang diminati oleh pengguna. Algoritma A* merupakan algoritma yang paling sering digunakan untuk menyelesaikan pencarian secara heuristic dengan menggunakan fungsi evaluasi untuk mengarahkan seleksi dari sekumpulan titik (Zhang and Zhao, 2014). Persamaan yang digunakan seperti pada persamaan (1) berikut

$$f(n) = g(n) + h(n) \quad (1)$$

Dimana:

$g(n)$ mewakili jarak total dari titik asal ke titik n , sedangkan $h(n)$ adalah fungsi heuristic yang menyatakan estimasi jarak dari titik n ke titik tujuan akhir.

Diagram alur dari algoritma A* ditunjukkan pada Gambar 6. A* diawali dengan menyiapkan dua variabel penampung yaitu *closedset* dan *openset*. *Closedset* berisi sekumpulan titik yang sudah dievaluasi, sedangkan *openset* berisi sekumpulan titik yang belum dievaluasi. Selain itu juga terdapat variabel penampung *total_path* yang menyimpan langkah urutan titik dengan jarak tempuh paling efisien. Dalam setiap perulangan titik di dalam *openset*, fungsi jarak selalu dihitung untuk setiap titik tetangga hingga dipilih nilai f yang paling kecil. Titik yang menghasilkan nilai f terkecil selanjutnya akan ditunjuk sebagai n dan mengulang kembali proses yang sama hingga tidak ditemukan lagi titik di dalam *openset*. Akhir dari alur proses menghasilkan *total_path* yang memuat urutan kunjungan titik-titik mulai dari titik awal hingga titik tujuan dalam jarak tempuh yang paling minimal. *Total_path* inilah yang nantinya membentuk rekomendasi rute yang ditampilkan sebagai hasil dari sistem.

5) Melakukan visualisasi rute.

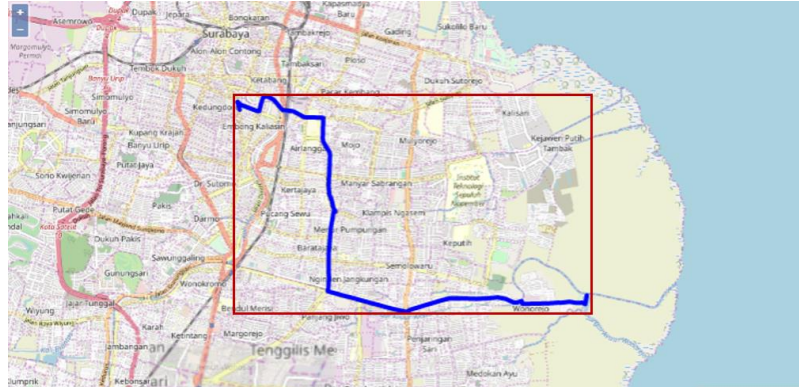
Tahap sebelumnya menghasilkan *total_path* yang berisi sekumpulan titik-titik *latitude* dan *longitude* dari tempat-tempat wisata yang akan dikunjungi dalam rute dalam format data bertipe GeoJSON.


```

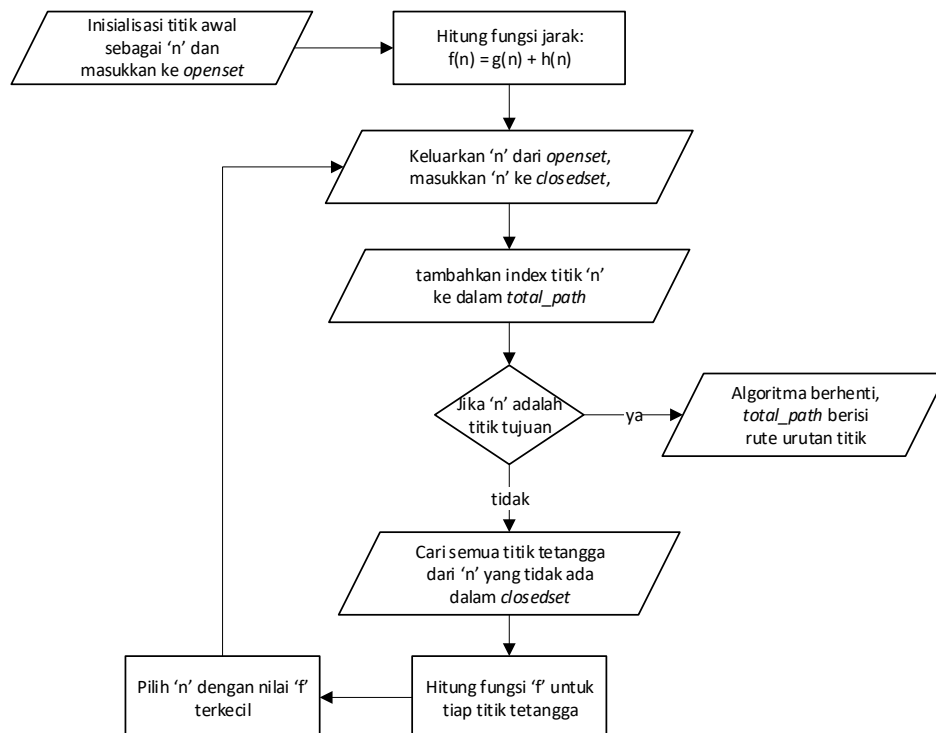
1. SELECT DISTINCT ?x0 ?x1 WHERE {
2. {?x0 rdfs:subClassOf* ta:""+input+"".
3. ?x0 rdfs:label ?x1.

```

Gambar 4. Sintaks SPARQL untuk Mendapatkan Tempat Wisata.



Gambar 5. Batasan Peta pada Rute Kunjungan Tempat Wisata



Gambar 6. Flowchart Algoritma A* untuk Mencari Urutan Kunjungan

6) Menampilkan hasil rekomendasi rute kepada pengguna.

Tahap terakhir adalah memvisualisasikan hasil rekomendasi rute bertipe GeoJSON agar dapat dimengerti oleh pengguna. Titik-titik koordinat tersebut dipetakan ke dalam peta dasar OSM dan ditampilkan di halaman utama aplikasi. Gambar 5 juga menunjukkan bentuk akhir dari visualisasi, dimana rekomendasi rute kunjungan wisata ditunjukkan oleh garis berwarna biru. Selain berupa peta, sistem juga menampilkan daftar nama-nama tempat wisata dari titik-titik lokasi wisata yang dilalui oleh garis rute, disertai dengan informasi tambahan mengenai jam buka dan jam tutup dari tempat wisata tersebut. Tampilan hasil akhir

semacam ini diharapkan dapat memberikan informasi yang lengkap dan memudahkan pengguna untuk mengetahui detail tempat wisata yang direkomendasikan oleh sistem.

HASIL

Uji coba yang dilakukan menggunakan batasan peta kota dan juga tempat wisata di Surabaya. Pengujian menerapkan tiga skenario yang berbeda dengan tujuan untuk mengujikan fungsionalitas sistem berdasarkan ragam kemungkinan masukan yang diberikan oleh pengguna. Skenario juga bermaksud mengujikan pengenalan makna kategori dan subkategori wisata oleh ontologi untuk menjawab kueri pengguna. Ketiga kategori tersebut akan dijabarkan sebagai berikut.

1. *Skenario Rute Tanpa Pilihan Kategori Wisata*

Pada skenario ini pengguna memilih titik awal dan titik tujuan rute tanpa memasukkan kategori wisata yang ingin dikunjungi. Skenario dimaksudkan untuk mengujikan hasil rute perjalanan yang diproduksi oleh algoritma A*. Sebagai pembanding, hasil visualisasi rute dicocokkan dengan hasil rute dari OSM dan Google Maps untuk menilai keakuratan rute. Pengguna memilih “Ekowisata Mangrove” sebagai titik awal dan “Tunjungan Plaza” sebagai titik tujuan. Gambar 7.a. merupakan hasil visualisasi dari sistem rekomendasi peneliti yang apabila dibandingkan dengan hasil rute skenario yang sama pada Google Maps dan OSM tidak banyak menunjukkan perbedaan bentuk rute. Hal ini dikarenakan dalam skenario ini pengguna tidak memasukkan kategori wisata apapun yang ingin dikunjungi.

2. *Skenario Rute dengan 1 Pilihan Kategori*

Skenario 2 menggunakan data masukan titik awal dan titik tujuan yang sama dengan Skenario 1. Namun pada skenario ini, pengguna ingin mengunjungi wisata “Pusat Perbelanjaan” yang bisa dihampiri di sepanjang rute perjalanan. Skenario ini juga ditujukan untuk mengujikan pemaknaan taksonomi kategori wisata pada sistem. Berdasarkan penjelasan pada bab sebelumnya, Pusat Perbelanjaan merupakan *super class* dari beberapa kategori lain seperti Mal dan Pasar Malam. Hasil rekomendasi rute yang dimunculkan oleh sistem juga memuat tempat wisata lain yang memiliki kategori Pusat Perbelanjaan beserta sub kategorinya.



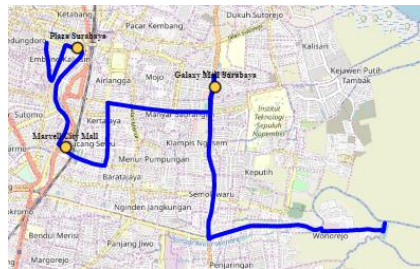
Gambar 7.a. Hasil Rute Skenario 1 pada Sistem



Gambar 7.b. Rute Skenario 1 pada Google Maps



Gambar 7.c. Rute Skenario 1 pada OSM



Gambar 8.a. Hasil Rute Skenario 2 pada Sistem



Gambar 9.a. Hasil Rute Skenario 3 pada Sistem

Rekomendasi Tujuan Wisata

Nama Lokasi	Alamat	Jam Buka	Jam Tutup
Galaxy Mall Surabaya	Jalan Dharmasusada Indah Timur No.35-37, Mulyorejo, Kota SBY, Jawa Timur 80115	10:00:00	22:00:00
Marvell City Mall	Jl. Ngagel No.123, Ngagel, Wonokromo, Kota SBY, Jawa Timur 60246	10:00:00	22:00:00
Plaza Surabaya	Jl. Pemuda No. 33-37, Surabaya, 60271	10:00:00	22:00:00

Gambar 8.b. Informasi pendukung dari tempat-tempat wisata Pusat Perbelanjaan yang dilalui oleh rute rekomendasi pada Skenario 2 (Gambar 8.a.).

Rekomendasi Tujuan Wisata

Nama Lokasi	Alamat	Jam Buka	Jam Tutup
Taman Apsari	Jl. Taman Apsari No.63, Embong Kaliasin, Genteng, Kota SBY, Jawa Timur 80271	null	null
Plaza Surabaya	Jl. Pemuda No. 33-37, Surabaya, 60271	10:00:00	22:00:00
Taman Persahabatan	Jl. Sulawesi No.87, Ngagel, Wonokromo, Kota SBY, Jawa Timur 60265	null	null
Gereja Cathedral	Jl. Polisi Istimewa, Keputran, Tegalsari, Kota SBY, Jawa Timur 60265	null	null
Marvell City Mall	Jl. Ngagel No.123, Ngagel, Wonokromo, Kota SBY, Jawa Timur 60246	10:00:00	22:00:00
Taman Flora	Jl. Raya Manyar No.80A, Baratajaya, Gubeng, Kota SBY, Jawa Timur 60284	08:00:00	17:00:00
Galaxy Mall Surabaya	Jalan Dharmasusada Indah Timur No.35-37, Mulyorejo, Kota SBY, Jawa Timur 80115	10:00:00	22:00:00
Taman Harmoni	Jl. Keputih Tegal Timur II No.249, Keputih, Kec. Sukolilo, Kota SBY, Jawa Timur 60111	06:00:00	18:00:00

Gambar 9.b. Informasi pendukung dari tempat-tempat wisata Pusat Perbelanjaan, Taman dan Wisata Religi yang dilalui oleh rute rekomendasi pada Skenario 3 (Gambar 9.a.)

Pada Gambar 8.a. rute melewati Marvel City Mall dan Plaza Surabaya yang merupakan wisata dengan kategori Mal. Hasil visualisasi rute memiliki alur yang sedikit berbeda bila dibandingkan dengan rute pada Gambar 7.a., hal ini dikarenakan sistem juga memperhitungkan keberadaan pusat perbelanjaan yang masih berada pada *bounding box* antara “Ekosistem Mangrove” dengan “Tunjungan Plaza”. Sedangkan Gambar 8.b. merupakan tampilan informasi pendukung dari rekomendasi tempat wisata yang bisa dikunjungi mengenai alamat dan jam buka.

3. Skenario Rute dengan lebih dari satu Kategori

Dengan data masukan titik awal dan titik tujuan yang masih sama dengan skenario-skenario sebelumnya, pada Skenario 3 pengguna menambahkan kategori pencarian wisata baru selain pusat perbelanjaan, yaitu “Taman” dan “Wisata Religi”. Skenario ini dimaksudkan untuk mengujicobakan kueri SPARQL apabila pengguna memasukkan kategori wisata lebih dari satu macam. Gambar 9.a. menampilkan bentuk rute rekomendasi yang dihasilkan oleh sistem. Rute memiliki alur kunjungan wisata yang lebih kompleks mengikuti ragam kategori wisata yang ingin dikunjungi oleh wisatawan. Semakin banyak kategori yang dimasukkan, maka rute juga akan semakin rumit dan panjang.

Hasil rekomendasi pada Gambar 9.a. menunjukkan bahwa rute melewati beberapa tempat wisata lain yang direpresentasikan dengan titik kuning pada garis biru. Sedangkan garis biru menandakan rute yang bisa dilewati. Dari Gambar 9.b. yang menampilkan informasi penunjang rekomendasi tempat wisata, dapat diketahui bahwa beberapa dari tempat tersebut merupakan *instance* dari kategori wisata “Taman”, seperti Taman Flora dan juga beberapa lainnya merupakan anggota dari kategori “Wisata Religi” seperti Gereja Cathedral.

Dari ketiga skenario ini dapat disimpulkan bahwa sistem rekomendasi rute kunjungan wisata yang dibangun telah memenuhi kebutuhan fungsionalitasnya dengan tepat.

PEMBAHASAN

Berdasarkan hasil pengujian tiga skenario yang ada pada bab sebelumnya, dapat disimpulkan bahwa secara fungsionalitas sistem telah dapat bekerja sesuai dengan tujuannya. Peneliti menggunakan keberadaan *bounding box* untuk membatasi hasil pencarian dan memastikan rekomendasi tempat wisata yang dihasilkan tidak terlampaui jauh dari rute sebenarnya. Keberadaan *bounding box* ini memungkinkan rute tidak memenuhi semua kategori wisata yang diminta oleh pengguna, dikarenakan tempat wisata yang ada dalam kategori tersebut terletak di luar *bounding box* sehingga dilibatkan dalam perhitungan rekomendasi.

Penggunaan ontologi terbukti mampu menjawab kebutuhan pencarian akan kategori wisata beserta subkategori-subkategori yang ada di dalamnya. Konsekuensi penerapan ontologi yang biasanya membutuhkan waktu proses yang lama dalam kasus ini tidak terjadi, dikarenakan peneliti melakukan pembatasan sumber daya dengan hanya memperhitungkan taksonomi *classes* saja. Namun, apabila kebutuhan penalaran ontologi sistem bertambah maka bisa dimungkinkan waktu proses yang diperlukan sistem juga ikut bertambah seperti pada penelitian-penelitian sebelumnya.

Berdasarkan waktu proses yang berhasil diamati dari beberapa kali uji coba pada ketiga skenario tersebut, dapat diambil rata-rata waktu proses untuk Skenario 1 (tanpa kategori wisata), Skenario 2, dan Skenario 3 secara berurutan adalah 1.58 ms, 4.85 ms, dan 6.29 ms. Ketiganya masih dalam rentang respon waktu yang cukup singkat dan wajar bagi pengguna, mengingat untuk aplikasi berbasis web rentang waktu respon di bawah 1 s akan dianggap wajar oleh pengguna sistem (*Response Time Limits: Article by Jakob Nielsen, no date*).

KESIMPULAN

Sistem rekomendasi rute kunjungan wisata yang dibangun telah berhasil menampilkan hasil rekomendasi yang sesuai dengan menerapkan solusi pemetaan peta tempat wisata dari OSM ke dalam basis data spasial, menerapkan A* untuk menentukan rute perjalanan wisata serta memanfaatkan ontologi untuk pengkategorian tempat wisata yang diperhitungkan pada penentuan rute kunjungan. Kueri SPARQL yang diterapkan

mampu menjawab kebutuhan pencarian tempat wisata dengan mempertimbangkan makna taksonomi dari kategorisasi tempat wisata yang tersimpan dalam basis data spasial PostgreSQL maupun skema file RDF.

SARAN

Pembangunan sistem rekomendasi yang dilakukan pada penelitian masih sangat awal dan inisial. Beberapa saran perbaikan dapat digunakan sebagai acuan dalam penelitian selanjutnya. Peneliti mempertimbangkan untuk menggunakan jenis algoritma pencarian lain selain A* untuk mencari rute perjalanan yang lebih optimal. Selain itu, keberadaan *bounding box* sebagai batasan pencarian rekomendasi tempat wisata perlu untuk dikaji ulang, mengingat beberapa tipe wisatawan mungkin menghendaki rute yang lebih fleksibel dalam jarak tempuh maupun waktu. Sistem juga dimungkinkan untuk dikembangkan dalam bentuk aplikasi perangkat bergerak yang diharapkan dapat meningkatkan kebergunaan sistem secara luas.

DAFTAR RUJUKAN

- Astrova, I., Korda, N. and Kalja, A. (2007) 'Storing OWL ontologies in SQL relational databases', *International Journal of Electrical, Computer and Systems Engineering*, 1(4), pp. 242–247.
- Batsakis, S. and Petrakis, E. G. M. (2010) 'SOWL: spatio-temporal representation, reasoning and querying over the semantic web', in. ACM, p. 15.
- Cui, S.-G., Wang, H. and Yang, L. (2012) 'A simulation study of A-star algorithm for robot path planning', in *16th international conference on mechatronics technology*, pp. 506–510.
- Daramola, O., Adigun, M. O. and Ayo, C. K. (2009) 'Building an ontology-based framework for tourism recommendation services', *Building an Ontology-Based Framework for Tourism Recommendation Services*, pp. 135–147.
- Delgado, J. A. and Davidson, R. (2002) *Knowledge bases and user profiling in travel and hospitality recommender systems*. Citeseer.
- Euzenat, J., Roşoiu, M.-E. and Trojahn, C. (2013) 'Ontology matching benchmarks: generation, stability, and discriminability', *Web Semantics: Science, Services and Agents on the World Wide Web*, 21, pp. 30–48.
- Lehmann, J. et al. (2015) 'DBpedia—a large-scale, multilingual knowledge base extracted from Wikipedia', *Semantic Web*, 6(2), pp. 167–195.
- Moreno, A. et al. (2013) 'SigTur/e-destination: ontology-based personalized recommendation of tourism and leisure activities', *Engineering applications of artificial intelligence*. Elsevier, 26(1), pp. 633–651.
- osm2pgrouting - Import OSM data into pgRouting Database — Open Source Routing Library (no date). Available at: <http://pgrouting.org/docs/tools/osm2pgrouting.html> (Accessed: 21 April 2020).
- Pazzani, M. J. (1999) 'A framework for collaborative, content-based and demographic filtering', *Artificial intelligence review*. Springer, 13(5–6), pp. 393–408.
- Response Time Limits: Article by Jakob Nielsen (no date). Available at: <https://www.nngroup.com/articles/response-times-3-important-limits/> (Accessed: 17 April 2020).
- Sebastia, L. et al. (2009) 'e-Tourism: a tourist recommendation and planning

- application', *International Journal on Artificial Intelligence Tools*. World Scientific, 18(05), pp. 717–738.
- Zhang, Z. and Zhao, Z. (2014) 'A multiple mobile robots path planning algorithm based on A-star and Dijkstra algorithm', *International Journal of Smart Home*, 8(3), pp. 75–86.