

Perancangan dan Pembuatan Perangkat Lunak Rekomendasi Pemilihan *Real Estate* dengan Metode *Mashup*

Mochamad Rizky, Dwi Sunaryono, Isye Ariesanti
Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)
Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia
e-mail: dwi@its-sby.edu

Abstrak—Pada studi ini dibuat suatu perangkat lunak untuk memberikan rekomendasi pemilihan *real estate*. Perangkat lunak ini dibuat untuk menyajikan informasi mengenai *real estate* secara lengkap, baik informasi karakteristik maupun informasi akses terhadap fasilitas umum. Informasi akan diambil dari *web* yang sudah tersedia. Informasi yang didapatkan tersebut akan diolah dan digabungkan dengan layanan *web* Google Maps API sehingga akan didapat informasi mengenai akses *real estate* terhadap fasilitas umum. Perangkat lunak ini dibuat dalam bentuk sistem informasi *web* agar mudah diakses oleh banyak orang. Perangkat lunak ini dibuat menggunakan teknik *mashup* dengan memanfaatkan teknologi HTML, ASP .NET, JavaScript, JQuery, AJAX, CSS 3, Google Maps JavaScript API, dan layanan *web* Google Maps. Dengan adanya perangkat lunak ini, diharapkan pembeli dan pengusaha *real estate* dapat memperoleh informasi lengkap serta optimal, sehingga dapat memberikan keuntungan yang besar. Disamping itu, dengan adanya aplikasi ini pengguna dapat dimudahkan dalam mencari *real estate* dengan fasilitas Google Maps.

Kata Kunci—*Mashup*, Pemilihan, Properti, Rekomendasi.

I. PENDAHULUAN

SEIRING dengan pesatnya laju pertumbuhan ekonomi di kota besar, menyebabkan meningkatnya kebutuhan akan *real estate* (tempat tinggal, pertokoan dan gedung perkantoran) di kota tersebut. Meningkatnya kebutuhan *real estate* di perkotaan merupakan salah satu aspek ekonomi yang sangat menguntungkan dan banyak diminati investor saat ini. Evaluasi dalam hal aset dan fasilitas yang terdapat pada *real estate* merupakan salah satu titik fokus yang harus diperhatikan dalam berinvestasi pada bidang ini. Hal tersebut bertujuan agar *real estate* tersebut dapat berkembang dan memberikan keuntungan yang besar bagi investor maupun pembeli.

Dalam memilih *real estate* yang baik dan menguntungkan, banyak sekali aspek yang harus diperhatikan, antara lain: karakteristik *real estate* (harga, luas tanah, luas bangunan, dan semua yang dimiliki oleh *real estate* tersebut), akses terhadap fasilitas umum (sekolah, rumah sakit, pasar, stasiun, dan fasilitas penunjang lain), dan kondisi lingkungan (keamanan, kesehatan, hubungan sosial). Dalam sebuah survei yang dilakukan oleh Giudice dari Universitas Naples menyimpulkan bahwa aspek yang paling diperhatikan oleh seseorang dalam membeli *real estate* adalah akses terhadap fasilitas umum dan kondisi lingkungan [1].

Saat ini telah banyak tersedia sistem informasi berupa situs *web* yang menyediakan informasi penjualan *real estate* di internet. Namun semua situs *web* tersebut hanya menyediakan informasi mengenai karakteristik *real estate* dan tidak menyediakan informasi mengenai akses terhadap fasilitas umum dan kondisi lingkungan. Sehingga pembeli ataupun investor tidak dapat secara optimal memilih *real estate* yang ideal. Agar pembeli dapat memilih *real estate* yang ideal, dibutuhkan sebuah sistem informasi yang menyediakan informasi secara lengkap mengenai *real estate* berupa akses *real estate* terhadap fasilitas umum dan kondisi lingkungan. Sehingga dapat memberikan rekomendasi pada pengguna mengenai *real estate* yang sesuai. Hal tersebut dapat dilakukan dengan memberikan informasi berupa peta lokasi yang disertai dengan informasi fasilitas umum dan karakteristik *real estate*.

Pada pembuatan makalah ini telah diimplementasikan sebuah sistem pemberian rekomendasi *real estate* dengan beberapa fitur, yaitu: pencarian berdasarkan karakteristik dan akses dengan fasilitas umum, melihat informasi *real estate*, melihat informasi fasilitas umum disekitar *real estate*, dan memilih *real estate*. Pengembangan sistem ini menggunakan kerangka kerja .NET 4.0 berbasis bahasa C#.

Pengujian dilakukan dengan menggunakan metode *blackbox*. Pengujian dilakukan pada seluruh fitur dengan membuat alur skenario aktivitas dan data tertentu untuk memastikan semua fitur berjalan semestinya.

II. METODOLOGI

A. Web Mashup

Web mashup adalah suatu situs *web* atau aplikasi *web* yang mengombinasikan dua atau lebih situs *web*. Situs *web* yang dihasilkan *mashup* merupakan situs *web* yang baru dan berbeda dari sumber aslinya. *Web mashup* biasanya digunakan untuk lebih memperkaya informasi dari sebuah situs *web*. Informasi yang dihasilkan bisa berasal dari HTML, XML, API ataupun layanan *web* [2]. Dengan menggunakan *web mashup*, aplikasi ini akan dapat menggabungkan informasi dari berbagai situs sumber dan menggabungkannya dengan layanan *web* Google Maps.

B. Google Maps

Google Maps merupakan salah satu layanan yang diberikan oleh Google untuk membantu pengguna internet dalam segala hal yang berhubungan dengan peta lokasi. Jasa layanan yang dapat digunakan langsung oleh

pengguna dapat diakses dalam situs <http://maps.google.com>. Data yang terdapat pada Google Maps diambil dari banyak satelit yang berada di seluruh dunia, sehingga dapat menyajikan data yang lengkap dan akurat [3].

Pada pembangunan aplikasi yang dijelaskan pada makalah ini, terdapat dua layanan dari Google yang digunakan yaitu Google Maps JavaScript API dan layanan web Google Maps [4]. Google Maps JavaScript API digunakan untuk menampilkan data rekomendasi pada peta yang terdapat pada bagian antarmuka. Sedangkan layanan web Google Maps digunakan untuk mendapat informasi peta lokasi mengenai suatu *real estate* dan informasi tempat lainnya.

Google Maps JavaScript API merupakan sebuah layanan yang disediakan oleh Google Maps dimana pengguna dapat menambahkan peta Google Maps pada situs web mereka. Selain itu Google Maps JavaScript API juga mengizinkan pengguna untuk menambahkan berbagai informasi di dalamnya. Sesuai dengan namanya, untuk dapat melakukan akses pada layanan ini digunakan bahasa pemrograman JavaScript.

Layanan web Google Maps merupakan layanan yang diberikan oleh Google Maps sebagai antarmuka yang dapat digunakan oleh pengguna untuk meminta berbagai data mengenai map API. Contoh data yang dapat didapat antara lain: koordinat tempat, jarak satu tempat dengan tempat lainnya, petunjuk jalan, dan mencari kemiringan antara dua tempat. Data tersebut nantinya dapat digunakan untuk aplikasi map yang dimiliki oleh pengguna. Layanan web Google Maps dibagi menjadi beberapa layanan web sesuai dengan penggunaannya yang terdapat pada Tabel 1.

Pada aplikasi ini hanya digunakan dua layanan web Google Maps untuk menunjang fungsionalitas aplikasi ini

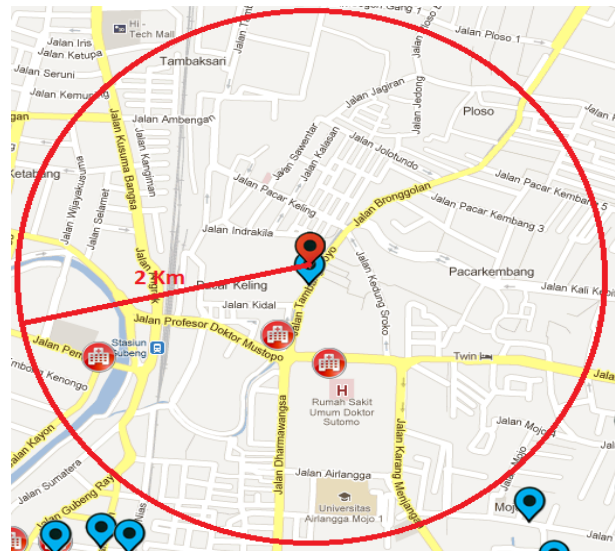
Tabel 1.

Jenis Layanan Google Maps

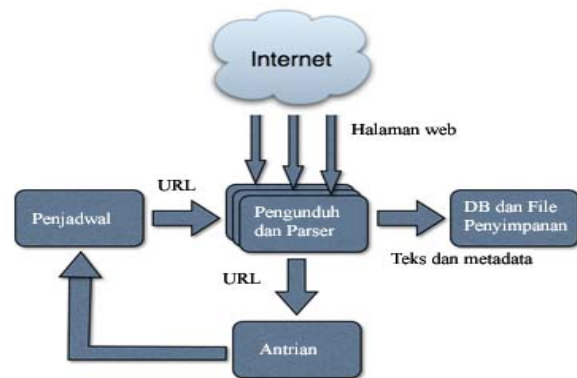
Layanan web	Fungsi Layanan
<i>Direction API</i>	Menyediakan informasi jarak perjalanan dan waktu perjalanan dari kumpulan asal menuju kumpulan tujuan [5].
<i>Distance Matrix API</i>	Menghitung arah antara beberapa lokasi dengan menggunakan permintaan HTTP [5].
<i>Elevation API</i>	Menyediakan informasi elevasi atau ketinggian untuk semua lokasi di permukaan bumi, termasuk juga kedalaman suatu lokasi di lautan.
<i>Geocoding API</i>	Merubah suatu alamat (seperti "Jalan Karang Menjangan, Surabaya, Indonesia") menjadi informasi geografis (seperti lintang -7.2739630 dan bujur 112.7481850).

yaitu *Distance Matrix API* dan *Geocoding API*. Selain itu, untuk dapat menyediakan informasi fasilitas umum digunakan layanan lain dari Google yaitu *Google Place*.

Google Place adalah layanan yang memberikan informasi mengenai sebuah tempat seperti perkantoran, lokasi geografis, atau tempat-tempat yang menarik dengan menggunakan permintaan HTTP. Permintaan yang dikirimkan pada *Google Place* mengenai suatu lokasi harus berupa koordinat lintang bujur [6]. Terdapat banyak layanan yang disediakan oleh *Google Place* namun dalam aplikasi rekomendasi *real estate* ini, *Google Place* hanya digunakan sebagai layanan mencari tempat dengan kriteria tertentu dan dalam radius tertentu. Penggunaan layanan ini dalam aplikasi ini dijelaskan pada Gambar 1. Pada gambar



Gambar. 1. Mencari Rumah Sakit dalam Radius 2 Km.

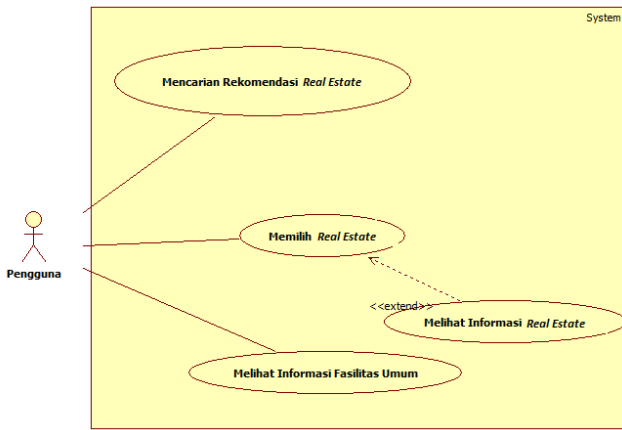


Gambar. 2. Arsitektur Web Crawler.

tersebut layanan *Google Place* bertugas untuk mencari fasilitas umum rumah sakit yang berada pada jarak radius 2 km.

C. Web Crawler

Web crawling adalah proses mengambil kumpulan halaman dari sebuah web untuk dilakukan pengindeksan sehingga menghasilkan informasi untuk aplikasi rekomendasi pemilihan *real estate* ini [7]. Pada Gambar 2 dijelaskan bagaimana aplikasi mengambil data tentang penjualan *real estate* dari sebuah situs web penjualan *real estate*. Aplikasi ini akan berjalan menelusuri halaman web dan mengumpulkan dokumen-dokumen atau data-data mengenai penjualan *real estate* yang ada di dalamnya. Selanjutnya *web crawler* akan membangun sebuah daftar indeks untuk memudahkan proses pencarian. Kemudian *web crawler* akan menganalisis halaman web yang telah tersimpan sebelumnya dengan cara mengindeks setiap kemungkinan *term* yang terdapat di dalamnya. Data *term* yang ditemukan disimpan dalam sebuah indeks basis data untuk digunakan dalam pencarian selanjutnya. Secara garis besar *web crawler* berfungsi untuk mengumpulkan, memilah dan menyimpan data untuk memberikan kemudahan dalam pengaksesan informasi secara tepat dan akurat. Tujuan dari penyimpanan data berupa indeks adalah untuk performansi dan kecepatan dalam menemukan informasi yang relevan berdasarkan masukan dari pengguna.



Gambar. 3. Diagram Kasus Penggunaan.

III. ANALISIS DAN PERANCANGAN

A. Perancangan Kasus Penggunaan

Secara garis besar perancangan perangkat lunak ini dijelaskan dengan menggunakan diagram kasus penggunaan yang ditunjukkan oleh Gambar 3. Dalam perancangan kasus penggunaan, hanya terdapat satu aktor yakni pengguna yang mengakses situs ini. Pengguna mampu melakukan beberapa hal diantaranya:

1. Meminta rekomendasi *real estate*.
2. Melihat informasi *real estate*.
3. Memilih *real estate*.
4. Melihat informasi fasilitas umum.

B. Perancangan Basis Data

Dalam perancangan basis data aplikasi ini, terdapat tujuh tabel. Tabel-tabel tersebut adalah sebagai berikut:

1. City (menyimpan nama kota dan koordinatnya).
2. Area (menyimpan data koordinat suatu area dalam satu kota).
3. RealEstate (menyimpan informasi mengenai *real estate* dari hasil *mashup* dan *crawler*).
4. RealEstateProperties (menyimpan data fasilitas umum yang dimiliki oleh *real estate* dan merupakan hasil normalisasi hubungan *many-to-many* antara tabel RealEstate dengan tabel PublikInfrastruktur).
5. PublikInfrastruktur (menyimpan informasi mengenai fasilitas umum).
6. InfrastrukturType (menyimpan tipe fasilitas umum).
7. UpdateTime (menyimpan waktu penyimpanan terakhir).

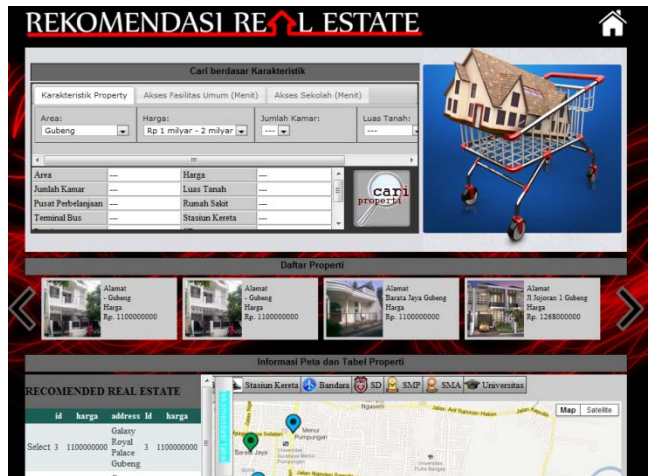
IV. IMPLEMENTASI

Implementasi perangkat lunak ini menggunakan bahasa pemrograman C# dengan kerangka kerja .NET 4.0 dan basis data SQL Server. Implementasi mencakup tiga hal yakni: implementasi antarmuka, implementasi basis data, dan implementasi kode program.

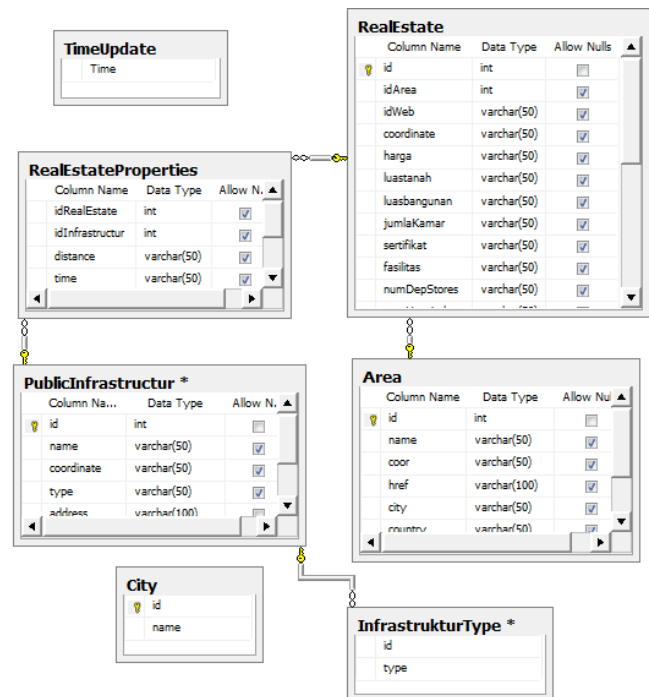
A. Contoh Implementasi Mencari Rekomendasi Properti

1) Implementasi Antarmuka

Gambar 4 merupakan antarmuka halaman mencari rekomendasi *real estate*. Sesuai dengan rancangannya, implementasi antarmuka meminta rekomendasi *real estate* memiliki data masukan berupa karakteristik *real estate* (area, harga, jumlah kamar, luas tanah) dan akses dengan



Gambar. 4. Implementasi Antar Mencari Rekomendasi Real Estate.



Gambar. 5. Gambar Diagram Basis Data.

fasilitas umum (jarak dengan pasar, jarak dengan rumah sakit dan jarak dengan fasilitas umum lainnya).

2) Implementasi Basis Data

Basis data aplikasi ini digambarkan oleh Gambar 5 yang merupakan diagram basis data hasil implementasi dari rancangan basis data sebelumnya. Tabel yang dihasilkan dari diagram tersebut antara lain RealEstate, Area, City, PublicInfrastruktur, RealEstateProperties, InfrastrukturType, dan UpdateTime.

3) Implementasi Kode Program

Implementasi kode program fitur mencari rekomendasi *real estate* melibatkan tiga kelas, yaitu: kelas *Model*, kelas *Controller*, dan kelas *View* [8].

Kelas *Model* terdiri dari kelas RealEstate, kelas PublicInfrastruktur, kelas RecomLogic, dan kelas InfrastrukturLogic. Kelas-kelas tersebut merupakan kelas-kelas yang berfungsi sebagai jembatan antara basis data dan kelas *Controller*. Kelas ini berisi aturan-aturan data yang boleh dimasukkan ke dalam basis data dan berisi perintah-perintah untuk melakukan *query* pada basis data.



Gambar. 6. Halaman Memasukkan Karakteristik.

Kelas *Controller* terdiri dari kelas *Recomendation*, kelas *HalamanProperty* dan kelas *HalamanInfrastruktur*. Kelas-kelas tersebut merupakan kelas-kelas di atas kelas *Model* yang telah disebutkan di atas. Kelas *Recomendation* berisi logika pemberian rekomendasi pada pengguna dan logika pengambilan data dari basis data. Sedangkan kelas *HalamanProperty* dan kelas *HalamanInfrastruktur* berfungsi mengatur informasi untuk ditampilkan pada kelas *View*.

Kelas *View* *HalamanUtama*, *HalamanRekomendasi*, *HalamanInfrastruktur* merupakan kelas yang berfungsi menampilkan hasil rekomendasi, informasi *real estate*, dan informasi fasilitas umum yang telah diberikan oleh *controller*. Informasi tersebut ditampilkan pada halaman HTML dan *Google Maps*.

V. PENGUJIAN

Pada bagian ini dibahas tentang rangkaian pengujian dan evaluasi. Pengujian dilakukan untuk mengetahui apakah fungsionalitas perangkat lunak sudah berjalan dengan benar. Pengujian dilakukan dengan menggunakan metode *blackbox*.

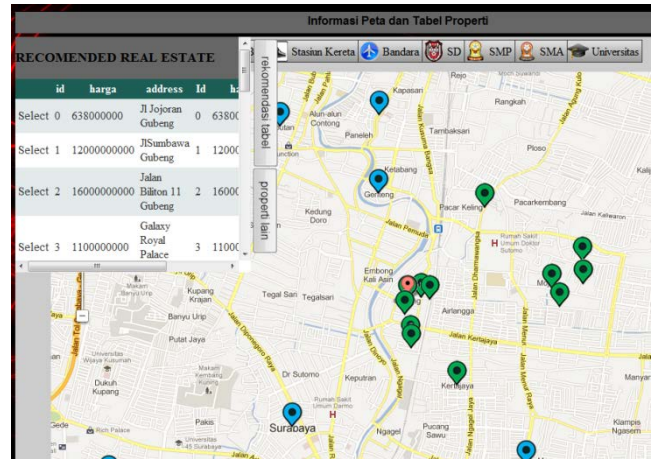
A. Contoh Pengujian Mencari Rekomendasi Properti

Skenario yang digunakan untuk mengetahui fungsionalitas fitur membuat album sebagai berikut:

1. Data masukan.
 - a. Karakteristik *real estate*.
 - b. Akses dengan fasilitas umum.
2. Data Uji
 - a. Data *real estate* di Surabaya.
 - b. Data fasilitas umum di Surabaya.
3. Prosedur
 - a. Pengguna memasukkan karakteristik *real estate*.
 - b. Pengguna memasukkan nilai kriteria akses dengan fasilitas umum.
 - c. Pengguna meminta rekomendasi dengan menekan tombol cari.
4. Hasil yang Diharapkan
 - a. Sistem memberikan hasil rekomendasi.
 - b. Situs *web* menampilkan data sesuai dengan rekomendasi.
5. Hasil yang Diperoleh
 - a. Sistem memberikan hasil rekomendasi.
 - b. Situs *web* menampilkan data sesuai dengan rekomendasi.
6. Kesimpulan

Proses meminta rekomendasi berhasil.

Gambar 6 menjelaskan langkah pengguna memasukkan kriteria *real estate*. Dan pada Gambar 7 menggambarkan



Gambar. 7. Halaman Data Hasil Rekomendasi.

Tabel 2. Hasil Pengujian Seluruh Fitur

Uji Coba	Hasil
Meminta rekomendasi.	OK
Memilih hasil rekomendasi.	OK
Memunculkan fasilitas umum	OK
Mengunduh informasi HTML dari situs sumber.	OK
Mengunduh dan mengolah data XML dari <i>Google Maps API</i> .	OK

Tabel 3. Hasil Pengujian Pengguna

Pertanyaan	Jumlah Tidak Setuju	Jumlah Tidak Tahu	Jumlah Setuju	Jumlah Setuju Sekali
<i>Website</i> Rekomendasi <i>Real Estate</i> tidak ditemukan <i>bug/error</i> selama digunakan.	1	3	8	3
Tampilan antarmuka dari Rekomendasi <i>Real Estate</i> cukup menarik dan mudah digunakan.	3	0	11	1
Fitur pencarian dan filter-filter yang ada pada Rekomendasi <i>Real Estate</i> membantu dalam menemukan <i>real estate</i> yang diinginkan.	0	2	12	1
Rekomendasi <i>real estate</i> yang diberikan Rekomendasi <i>Real Estate</i> sesuai dengan karakteristik yang dimasukkan pengguna.	0	5	9	1
Waktu untuk mendapatkan hasil cari atau rekomendasi dari Rekomendasi <i>Real Estate</i> terbilang cepat.	0	3	10	2
Situs Rekomendasi <i>Real Estate</i> sudah siap dikenalkan kepada masyarakat luas.	0	4	10	1

hasil yang didapat sesuai dengan masukan yang ada pada Gambar 6.

B. Hasil Pengujian Keseluruhan Fitur

Pada Tabel 2 akan ditunjukkan hasil dari semua uji fungsionalitas perangkat lunak.

C. Hasil Pengujian Pengguna

Pengujian pengguna merupakan pengujian untuk mengetahui kepuasan terhadap perangkat lunak ini. Pada Tabel 3 ditunjukkan rekapitulasi uji pengguna yang dilakukan oleh 15 responden.

VI. KESIMPULAN

Berdasarkan hasil pengamatan selama proses perancangan, implementasi, dan pengujian perangkat lunak ini, dapat diambil kesimpulan sebagai berikut:

- 1) Aplikasi yang dibangun dapat melakukan *mashup* antar situs sumber (rumah123.com) dan layanan Google Maps.
- 2) Berdasarkan hasil pengujian menunjukkan aplikasi yang dibangun dapat memberikan rekomendasi pembelian *real estate*.
- 3) Metode *mashup* antar situs pencarian *real estate* dan Google Maps dapat menjadi solusi dalam memberikan rekomendasi pencarian *real estate*.

UCAPAN TERIMA KASIH

Para penulis mengucapkan terima kasih kepada Allah SWT atas limpahan rahmat dan hidayah-Nya dan terima kasih kepada semua pihak yang telah membantu dalam penulisan makalah ini.

DAFTAR PUSTAKA

- [1] V. Giudice, *Decision Support System for Real estate Investment Choice*, University of Naples Italy (2009).
- [2] Mashups: "The new breed of Web app", (2006) <http://www.ibm.com/developerworks/xml/library/x-mashups/index.HTML>, diakses tanggal 3 Juni 2012.
- [3] Google Maps, "Google Maps JavaScript API v3", (2012) <https://developers.google.com/maps/documentation/javascript/?hl=id>, diakses tanggal 3 Juni 2012.
- [4] Google Maps, "The Google Geocoding API" <https://developers.google.com/maps/documentation/geocoding/?hl=id>, diakses tanggal 4 Juni 2012.
- [5] Google Maps, "The Google Distance Matrix API" <https://developers.google.com/maps/documentation/distancematrix/?hl=id>, diakses tanggal 4 Juni 2012.
- [6] Google Maps, "The Google Places API", (2012), <https://developers.google.com/places/documentation/>, diakses tanggal 9 Juni 2012.
- [7] Jones, M Tim, "Build a Web Spider", (2006), <http://www.ibm.com/developerworks/linux/library/l-spider/>, diakses tanggal 10 Juni 2012.
- [8] Microsoft ASP.NET Team, "ASP.NET MVC Overview", (Januari 2009), <http://www.asp.net/mvc/tutorials/older-versions/overview/asp-net-mvc-overview#author-info>, diakses tanggal 2 Juni 2012.