

## PEMBANGUNAN APLIKASI TRAVEL RECOMMENDER DENGAN METODE CASE BASE REASONING

Uung Ungkawa<sup>[1]</sup>, Dewi Rosmala<sup>[2]</sup>, Fanny Aryanti<sup>[3]</sup>

Jurusan Teknik Informatika  
Institut Teknologi Nasional Bandung

[uung@itenas.ac.id](mailto:uung@itenas.ac.id), [d\\_rosmala@itenas.ac.id](mailto:d_rosmala@itenas.ac.id)

### ABSTRAK

*Penalaran Berbasis Kasus / Case Based Reasoning (CBR) merupakan metode yang dipergunakan untuk membangun sebuah sistem berbasis pengetahuan. Sumber pengetahuan sistem diperoleh dengan mengumpulkan penanganan kasus-kasus oleh seorang ahli/ pakar. Keuntungan dengan penerapan metode ini adalah pembangun pengetahuan tidak perlu melakukan akuisisi pengetahuan secara langsung dengan seorang pakar. Pada penelitian ini diimplementasikan metode CBR untuk membantu perekomendasi wisata. Kasus-kasus yang dipergunakan dalam sistem diperoleh dari beberapa sumber data real tentang objek wisata di Jawa Barat. Sistem akan memberikan keluaran berupa jenis objek wisata yang direkomendasikan serta perkiraan biaya akomodasi yang didasarkan pada kemiripan kasus baru dengan pengetahuan yang dimiliki sistem.*

**Kata Kunci :** *Case Based Reasoning, Sistem Berbasis Pengetahuan, Travel Recommender*

### ABSTRACT

*Case-Based Reasoning (CBR) is a method used to build a knowledge-based system. Knowledge system management source is obtained by collecting the cases by an expert/ specialist. The advantage of this method is the application of knowledge builders do not have to do directly with the acquisition of knowledge of an expert. In this study, the authors implement a method of CBR to help tour recommendation. Case used in the system is obtained from multiple sources of data about the real attractions in West Java. The system will provide the output of the type recommended attractions and accommodation cost estimates are based on the similarity of a new case with the knowledge system.*

**Keywords:** *Case-Based Reasoning, Knowledge Based System, Travel Recomender*

## Latar Belakang

*Travelling* atau berwisata merupakan kegiatan yang saat ini sangat digemari oleh seluruh masyarakat Indonesia. Jawa Barat merupakan salah satu daerah di Indonesia yang memiliki banyak tempat wisata, terutama di kota Bandung. Jawa Barat merupakan salah satu tempat wisata yang disenangi masyarakat, hal ini bisa dilihat dengan tingkat kemacetan yang meningkat disetiap *week-end* dan *long week-end*. Banyak dibangunnya mal-mal, tempat wisata, rumah makan (*café / resto*), hotel-hotel dan factory outlet juga menunjukkan tingkat ketertarikan masyarakat terhadap kota Bandung, cuaca yang sejuk juga menjadi alasan para wisatawan memilih Bandung sebagai tempat berwisata.

Para wisatawan terkadang bingung menentukan lokasi wisata, hotel yang harganya terjangkau, kendaraan/ alat transportasi yang sebaiknya mereka gunakan dan biaya yang sebaiknya mereka persiapkan sebelum berpergian.

Sebuah aplikasi “*travel recommender*” yaitu sebuah aplikasi yang diciptakan untuk memberikan rekomendasi untuk para calon wisatawan. Aplikasi ini akan bermanfaat untuk calon wisatawan agar bisa mempersiapkan perjalanannya secara matang dan lebih memanfaatkan waktu mereka selama berlibur. Selain itu juga aplikasi rekomendasi travel seperti ini belum pernah dibuat khusus untuk daerah Jawa Barat.

*Case Base Reasoning* (CBR) memiliki kemampuan diagnosa berbasis kasus dan memberikan informasi secara otomatis berdasarkan pengetahuan terdahulu yang dapat direvisi untuk menyesuaikan dengan permasalahan terbaru. Sehingga pengetahuan CBR akan terus berkembang. Pemecahan masalah baru pada CBR dilakukan dengan cara mencari permasalahan sejenis di masa lampau dan memberikan solusi berdasarkan permasalahan yang paling mirip yang ada di dalam *case memory*. Permasalahan yang

dapat digunakan untuk memecahkan masalah tersimpan di dalam *case memory* yang dapat direvisi untuk memecahkan permasalahan di masa datang.

## Rumusan Masalah

Berdasarkan latar belakang maka dapat dirumuskan beberapa masalah sebagai berikut:

- Bagaimana mengimplementasikan *Case Base Reasoning* di dalam aplikasi *Travel Recommender*
- Bagaimana membangun suatu aplikasi yang pintar yang diharapkan dapat memberikan solusi/masukan terhadap rencana wisata yang terbaik seperti layaknya bertanya kepada seorang ahli.

## Tujuan

Tujuan dalam penelitian ini adalah membangun aplikasi *Travel Recommender* untuk calon wisatawan yang berbasis web dengan menggunakan metoda *case base reasoning* dan menggunakan bahasa pemrograman *PHP+jQuery* sebagai pengganti dari sistem manual dan dapat digunakan oleh user sebagai alat untuk mencari rekomendasi tentang objek wisata yang ada di Jawa Barat.

## Batasan Masalah

Adapun batasan masalah pada penelitian ini adalah sebagai berikut:

- Studi kasus dan objek yang diteliti dalam penelitian tugas akhir ini adalah objek wisata yang dikelola oleh Dinas Pariwisata Jawa Barat.
- Adapun item (variable) atau informasi rekomendasi yaitu jenis wisata, daerah wisata, alat transportasi yang digunakan, harga hotel dan peta lokasi tempat.
- Pembobotan nilai similaritas dibatasi dengan nilai 0.9, 0.8, 0.7

## Landasan Teori

### Sistem Rekomendasi<sup>[1]</sup>

Sistem rekomendasi merupakan suatu aplikasi untuk menyediakan dan merekomendasikan suatu item dalam membuat suatu keputusan yang diinginkan oleh pengguna.

Sistem rekomendasi pariwisata atau perjalanan menggunakan konsep ini untuk membantu para wisatawan untuk memutuskan tujuan perjalanan mereka, akomodasi dan peta lokasi tempat tujuan wisatawan.

### Case-Based Reasoning<sup>[3]</sup>

*Case-Based Reasoning* (CBR) adalah metode untuk menyelesaikan masalah dengan mengingat kejadian-kejadian yang sama/sejenis (*similar*) yang pernah terjadi di masa lalu kemudian menggunakan pengetahuan/ informasi tersebut untuk menyelesaikan masalah yang baru, atau dengan kata lain menyelesaikan masalah dengan mengadaptasi solusi-solusi yang pernah digunakan di masa lalu. Menurut Aamodt dan Plaza (1994) *Case-Based Reasoning* adalah suatu pendekatan untuk menyelesaikan suatu permasalahan (*problem solving*) berdasarkan solusi dari permasalahan sebelumnya.

*Case-based Reasoning* ini merupakan suatu paradigma pemecahan masalah yang banyak mendapat pengakuan yang pada dasarnya berbeda dari pendekatan utama AI lainnya. Suatu masalah baru dipecahkan dengan menemukan kasus yang serupa di masa lampau, dan menggunakannya kembali pada situasi masalah yang baru. Perbedaan lain dari CBR yang tidak kalah penting adalah CBR juga merupakan suatu pendekatan ke arah *incremental* yaitu pembelajaran yang terus-menerus.

Dalam *Case-Based Reasoning* ada empat tahapan yang meliputi:

#### a. *Retrieve*

Mendapatkan kembali kasus yang paling relevan (*similar*) dengan kasus yang baru. Tahap *retrieval* ini dimulai dengan menggambarkan sebagian masalah, dan diakhiri jika ditemukan kecocokan terhadap masalah sebelumnya yang tingkat kecocokannya paling tinggi. Bagian ini mengacu pada segi identifikasi, kecocokan awal, pencarian dan pemilihan serta eksekusi.

#### b. *Reuse*

Memodelkan/menggunakan kembali pengetahuan dan informasi kasus lama berdasarkan bobot kemiripan yang paling relevan ke dalam kasus yang baru, sehingga menghasilkan usulan solusi dimana mungkin diperlukan suatu adaptasi dengan masalah yang baru tersebut.

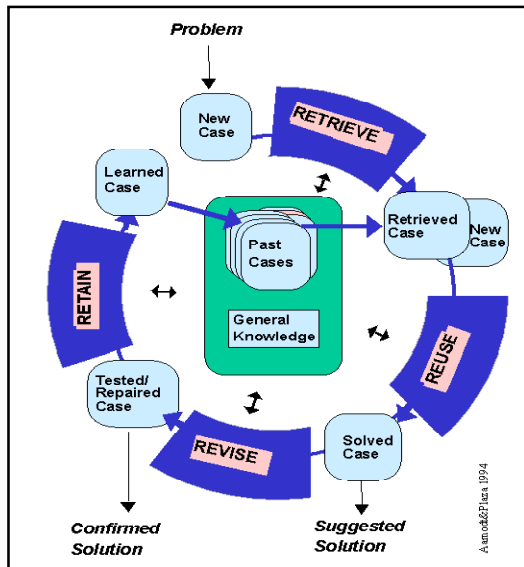
#### c. *Revise*

Meninjau kembali solusi yang diusulkan kemudian mengetesnya pada kasus nyata (simulasi) dan jika diperlukan memperbaiki solusi tersebut agar cocok dengan kasus yang baru.

#### d. *Retain*

Mengintegrasikan kasus baru yang telah berhasil mendapatkan solusi agar dapat digunakan oleh kasus-kasus selanjutnya yang mirip dengan kasus tersebut, tetapi jika solusi baru tersebut gagal, maka menjelaskan kegagalannya, memperbaiki solusi yang digunakan, dan mengujinya lagi.

Empat proses masing-masing melibatkan sejumlah langkah-langkah spesifik, yang dijelaskan pada Gambar 1



Gambar 1. Siklus Metode Case-Based Reasoning (Aamodt & Plaza, 1994)

Pada saat terjadi permasalahan baru, pertama-tama sistem melakukan proses *retrieve*. Proses *retrieve* melakukan dua langkah pemrosesan, yaitu pengenalan masalah dan pencarian persamaan masalah pada database.

Setelah proses *retrieve* selesai dilakukan, selanjutnya sistem melakukan proses *reuse*. Di dalam proses *reuse*, sistem menggunakan informasi permasalahan sebelumnya yang memiliki kesamaan untuk menyelesaikan permasalahan yang baru. Pada proses *reuse* dilakukan penyalinan, penyeleksian, dan melengkapi informasi yang digunakan. Selanjutnya pada proses *revise*, informasi tersebut dikalkulasi, dievaluasi, dan diperbaiki kembali untuk mengatasi kesalahan-kesalahan yang terjadi pada permasalahan baru.

Pada proses terakhir, sistem melakukan proses *retain*. Proses *retain* mengindeks, mengintegrasikan, dan mengekstrak solusi yang baru tersebut kedalam database.

Selanjutnya, solusi baru itu disimpan di dalam basis pengetahuan (*knowledge-base*) untuk menyelesaikan permasalahan

yang akan datang. Tentunya, permasalahan yang memiliki kesamaan.

(*k-nearest neighbor* atau *k-NN*)<sup>[4]</sup>

Algoritma *Nearest Neighbor Retrieval* (*k-nearest neighbor* atau *k-NN*) sebuah algoritma untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Kasus khusus di mana klasifikasi diprediksikan berdasarkan data pembelajaran yang paling dekat (dengan kata lain,  $k = 1$ ) disebut algoritma *nearest neighbor*.

Algoritma *nearest neighbor* berdasarkan pada proses pembelajaran menggunakan analogi / *learning by analogi*. *Training* sampelnya dideskripsikan dalam bentuk atribut numerik  $n$ -dimensi. Tiap sampel mewakili sebuah titik pada ruang  $n$ -dimensi. Dengan cara ini, semua *training* sampel disimpan pada pola ruang  $n$ -dimensi. Ketika diberikan "*unknown*" sampel, *k-nearest neighbor classifier* mencari pola ruang  $K$  *training* sampel yang paling dekat "*unknown*" sampel tersebut.  $K$  *training* sampel ini adalah  $k$  *nearestneighbor* dari *unknown* sampel. *Unknown* sampel ditetapkan dengan class yang paling umum diantara  $k$  *nearest neighbors*nya. Ketika  $k = 1$ , *unknown* sampel ditetapkan dengan class dari *training* sampel yang paling dekat dengan pola ruangnya.

Algoritma *nearest neighbor retrieval* menyimpan semua *training* sampel dan tidak

Membangun *classifier* sampai sampel baru (*unlabeled*) perlu diklasifikasikan, sehingga algoritma *nearest neighbor retrieval* sering disebut dengan *instance-based* atau *lazy learners*.

Rumus untuk menghitung bobot kemiripan (*similarity*) dengan *nearest neighbor*

*Retrieval* adalah:

$$\text{Similarity (problem, case)} = \frac{s_1 * w_1 + s_2 * w_2 + \dots + s_n * w_n}{w_1 + w_2 + \dots + w_n}$$

Keterangan:

S = *similarity* (nilai kemiripan)

W = *weight* (bobot yang diberikan)

### Analisis sistem

Dalam pengembangan aplikasi sistem rekomendasi wisata, sistem membutuhkan data real tentang tempat wisata, biaya dan lokasi hotel yang ada di Jawa Barat, selain itu juga dibutuhkan sebuah data hasil wawancara terhadap beberapa sample user untuk menentukan tingkatan dalam pembobotan hal yang biasanya didahulukan user ketika akan berwisata, yang digunakan dalam pembobotan dalam proses penghitungan similaritas untuk agar sistem dapat menghasilkan rekomendasi yang terbaik (Tabel 1).

### Tabel Sample Data Real

Tabel sample data real adalah sebuah kumpulan data real tentang lokasi wisata, daerah wisata, harga hotel maupun penginapan yang ada di Jawa Barat.

### Tabel Hasil Wawancara

Tabel hasil wawancara adalah kumpulan data dari hasil wawancara yang dilakukan untuk membuat pembobotan terhadap hal-hal yang telah ditentukan untuk pemberian rekomendasi wisata. Hal-hal tersebut adalah jenis wisata, daerah wisata dan akomodasi. Tabel 1 adalah hasil wawancara yang dilakukan.

Nilai bobot ini diberikan sebagai gambaran seberapa penting index tersebut dalam penentuan solusi dari kasus yang sama. Index yang dimiliki oleh tiap-tiap kasus adalah:

- Jenis Wisata
- Daerah Wisata
- Akomodasi

Sebagai contoh didalam penelitian ini jenis wisata ditentukan memiliki nilai bobot 0.9 yang artinya adalah index ini sangat menentukan jenis rekomendasi yang diberikan untuk kasus yang dicari, sedangkan untuk Daerah Wisata dan akomodasi diberi bobot 0.8 dan 0.7 yang artinya index ini tidak terlalu memberikan pengaruh yang signifikan.

### Gambaran singkat proses *case based reasoning*

Metode *Cased Based Reasoning* merupakan metode yang menerapkan 4 tahapan proses, yaitu proses *retrieve*, *reuse*, *revise*, dan *retain*. Cara kerja sistem secara umum berpedoman pada basis pengetahuan yang dimiliki oleh sistem yang bersumber dari data real untuk rekomendasi wisata di daerah Jawa Barat (Gambar 2) yang kemudian dihitung tingkat kemiripannya dengan kasus baru yang dimasukan pengguna (Gambar 3). Berdasarkan tingkat kemiripan kasus inilah system mengeluarkan rekomendasi wisata yang sesuai dengan rencana berwisata. *User* yang menginputkan rencana wisata.

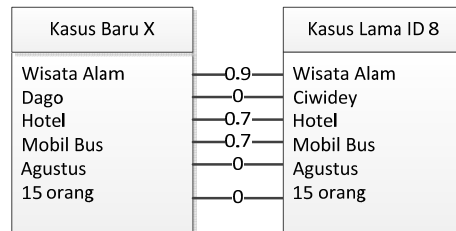
Rencana Wisata : ID 1 Jenis Wisata : Wisata Alam Daerah Wisata : Ciwidey Jumlah orang : 10 Waktu berwisata : Agustus Alat Transportasi : Mobil Bus Akomodasi : Hotel	Rencana Wisata : ID 2 Jenis Wisata : Wisata Alam Daerah Wisata : Lembang Jumlah orang : 10 Waktu berwisata : Agustus Alat Transportasi : Mobil Akomodasi : Villa
Rekomendasi Wisata ID 8 Jenis Wisata : Wisata Alam Daerah Wisata: Ciwidey Jumlah Orang : 10 Alat Transportasi : Mobil Bus Biaya Akomodasi : 3.000.000	Rekomendasi Wisata ID 15 Jenis Wisata : Wisata Alam Daerah Wisata: Lembang Jumlah Orang : 5 Alat Transportasi : Mobil Biaya Akomodasi : 500.000

Gambar 2. Contoh kasus rencana wisata

# JURNAL INFORMATIKA

Tabel 1 Data hasil Wawancara

No	Nama	Pekerjaan	Daerah Asal	Bobot pilihan hasil wawancara			Total
				Jenis Wisata	Daerah Wisata	Biaya Akomodasi	
1	Clarissa Octavianty	Mahasiswa	Rancaherang, Bandung	40%	30%	30%	100%
2	Danti Julianty	Mahasiswa	Bogor	30%	50%	20%	100%
3	Novi Yuliani	Pegawai Swasta	Rancaherang, Bandung	30%	50%	20%	100%
4	Ivan Rochim	Pegawai Swasta	Sarijadi, Bandung	50%	45%	5%	100%
5	Dadang Chandra	Pegawai Negri	Cimanggu	50%	50%	0%	100%
6	Annisa	Pegawai Swasta	Gegerkalong, Bandung	30%	30%	40%	100%
7	Hendra Leo	Pegawai Swasta	Serang	70%	20%	10%	100%
8	Irina Pramesty	Mahasiswa		50%	10%	40 %	100%
9	Arie Erlangga	Mahasiswa	Margahayu, Bandung	30%	20%	50%	100%
10	Iqbal Mutakin	Mahasiswa	Padalarang	40%	40%	20%	100%
	Rata-rata			42%	34.5%	23.5%	100%



Gambar 3. Contoh kasus baru

Similarity (X,8)

$$[(1*0.9)+(0*0.8)+(1*0.7)+(1*0.7)+(1*0)+(1*0)]$$

$$0.9 + 0.8 + 0.7 + 0.7 + 0 + 0$$

$$= 0.7419$$

**Perhitungan kasus 2:**

Bobot rencana wisata2 :

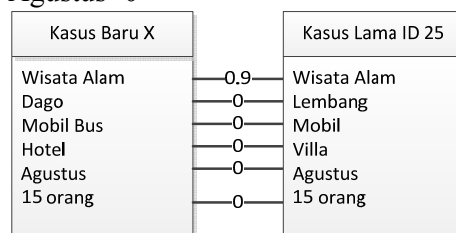
Wisata Alam = 0.9

Dago = 0.8

Hotel= 0.7

20 orang =0

Agustus=0



Bobot parameter (w):

JenisWisata = 0.9

LokasiWisata = 0.8

Akomodasi = 0.7

**Similarity (problem,case) =**

$$\frac{s1*w1+ s2*w2+ ..... + sn*wn}{w1 + w2 + ..... + wn}$$

Keterangan:

S=*similarity* (nilai kemiripan) yaitu 1 (sama) dan 0 (beda)

W = *weight* (bobot yang diberikan)

**Perhitungan kasus 1:**

Bobot rencana wisata1:

Wisata Alam = 0.9

Dago = 0.8

Hotel = 0.7

Mobil bus = 0.7

15 orang =0

Agustus=0

[Type text]



Similarity (X,25)

$$\frac{[(1*0.9)+(0*0.8)+(1*0)+(1*0)+(0*0.7)+(0*0.7)]}{0.9 + 0.8 + 0 + 0 + 0.7 + 0.7}$$

$$= 0.2903$$

### b. Proses Reuse

Dari perhitungan diatas kasus yang memiliki bobot kemiripan paling rendah adalahkasus ID 25 yaitu sebesar 0,2903. Kasus ID 8 menghasilkan bobot 0,7419

Pada proses *reuse*, solusi yang diberikan adalah solusi dengan bobot kemiripan kasus lama dengan kasus baru yang paling tinggi, dalam contoh kasus ini adalah kasus ID 8. Hasil perhitungan dengan bobot menunjukkan tingkat kepercayaan lebih dari 70%, jadi solusi kasus ID 8 lah yang direkomendasikan kepada pengguna yaitu Jenis Wisata: Wisata Alam, Daerah: Ciwidey, Alat Transportasi: Mobil Bus, Jumlah 15 orang, Biaya Akomodasi: 4.500.000,00.

### c. Proses Revise

Pada contoh ini kasus ID 8 sudah menghasilkan solusi dengan tingkat kepercayaan diatas 70%, jadi rekomendasi yang dihasilkan dapat langsung diberikan.

Tetapi jika ternyata setelah dilakukan proses perhitungan dan tidak ada kasus yang mirip dengan kasus baru tersebut maka dilakukan proses *revise*. Informasi berupa masukan rencana pada kasus baru yang tidak ditemukan kemiripannya dengan basis pengetahuan (*rule*) tersebut ditampung pada suatu tabel khusus (tabel *revise*) yang selanjutnya akan dievaluasi dan diperbaiki kembali oleh pakar untuk menemukan solusi yang tepat.

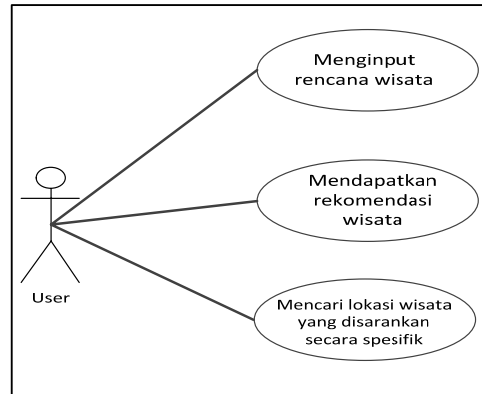
### d. Proses Retain

Setelah proses *revise* selesai dan sudah ditemukan solusi yang benar-benar tepat barulah pakar mulai menambah aturan dengan memasukkan data kasus baru yang sudah ditemukan solusinya tersebut ke dalam basis pengetahuan yang nantinya dapat digunakan untuk kasus berikutnya

yang memiliki permasalahan yang sama. Proses inilah yang disebut dengan proses *retain*.

### Use Case Diagram

Berikut merupakan gambaran proses pengembangan sistem yang dimodelkan dalam bentuk *use case*.



Gambar 4 Use Case Diagram sistem

Gambar 4 merupakan diagram *use case* dari proses utama yang terjadi pada pengembangan sistem aplikasi.

### Skenario Use Case

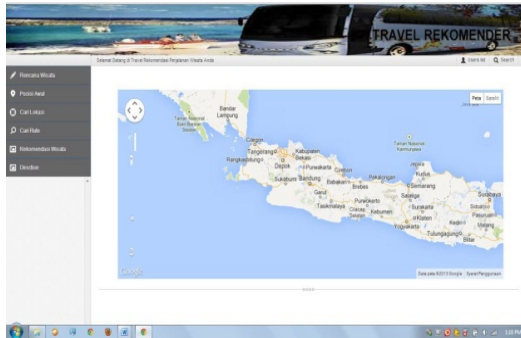
Untuk memberikan kejelasan dalam proses logik didalam melaksanakan fungsionalitas-fungsionalitas yang harus difasilitasi oleh sistem yang dinyatakan pada *use case diagram*, maka dibuatlah skenario *use case* untuk proses use case aktivitas pemberian rekomendasi wisata

### Implementasi Sistem

Impelementasi bertujuan untuk mengkonfirmasi modul – modul perancangan, sehingga pengguna dapat memberikan masukan kepada pengembang sistem.

### Halaman Utama Aplikasi

Halaman ini merupakan halaman utama pada aplikasi yang dibuat di *platform* android yang terdiri dari beberapa *button* (menggunakan *icon* android) sebagai simbol peralatan listrik yang dikendalikan. Berikut tampilan utama aplikasi pada Gambar 5.



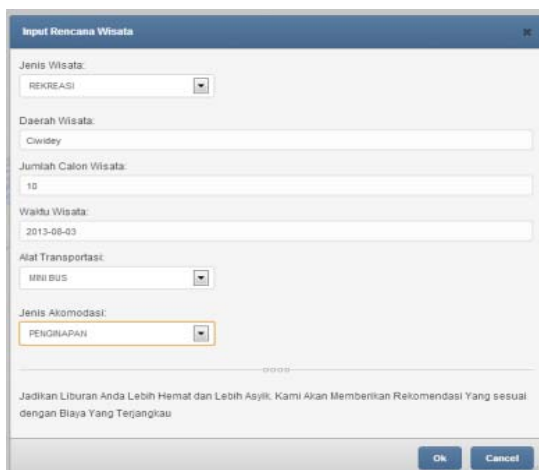
Gambar 5 Halaman awal aplikasi

## Pengujian Sistem

Pengujian aplikasi sistem rekomendasi wisata baru mencapai tahap pengujian proses uji *alpha*. Pengujian sistem ini baru dilakukan oleh pihak *developer* dan belum dilakukan oleh pihak *user* yang sebenarnya yaitu konsumen. Pada pengujian sistem ini dilakukan *black box testing* yang berfungsi untuk menguji setiap fungsi pada aplikasi yang telah dibuat.

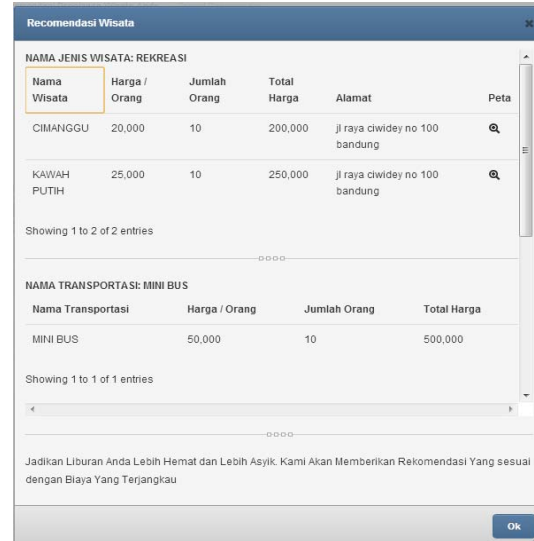
## Pengujian perekomendasi wisata

Pada tahap pengujian proses perekomendasi wisata ini sebelumnya dilakukan pengujian input rencana wisata terlebih dahulu, *user* harus melakukan input rencana wisata terlebih dahulu yaitu *user* memasukkan rencana wisata.. Tampilan halaman input rencana wisata yang harus dilengkapi *user* dapat dilihat pada Gambar 6.



Gambar 6 Tampilan halaman input rencana wisata

Selanjutnya adalah fungsi hasil rekomendasi wisata yang diberikan oleh sistem. Tampilan hasil pengujian fungsi hasil pemberian rekomendasi wisata dapat dilihat pada Gambar 7.



Gambar 7 Tampilan halaman rekomendasi wisata

Untuk menguji keberhasilan aplikasi *Travel Recommender* memberikan rekomendasi wisata, dilakukan percobaan dengan user menginputkan rencana wisatanya dipaparkan pada Tabel 1 pengujian rencana wisata dan tabel hasil perekomendasi wisata berdasarkan rencana wisata ditampilkan dalam Tabel 2 Hasil rekomendasi wisata.

Berdasarkan hasil pengujian yang dilakukan pengguna terhadap butir uji **input rencana wisata** dan terhadap butir uji **hasil perekomendasi wisata oleh sistem**, serta hasil pengujian keberhasilan perekomendasi wisata dapat disimpulkan bahwa tingkat keakuratan dari perekomendasi wisata yang dilakukan system perekomendasi wisata yang dilakukan sistem memiliki tingkat keberhasilan yang baik, terlihat dari setiap hasil percobaan memiliki hasil rekomendasi.



# JURNAL INFORMATIKA

Tabel 1 Hasil pengujian rencana

No	Jenis Wisata	Daerah Wisata	Calon Wisata	Waktu	Alat Transportasi	Jenis Akomodasi	Baik	Buruk
1	Wisata Alam	Lembang	15	Agustus	Mobil Bus	Villa	√	
2	Petualangan	Ciwidey	5	Juni	Mobil	Hotel	√	
3	Kuliner	Dago	10	Juli	Mobil	Hotel	√	
4	Wisata Alam	Dago	4	Juni	Motor	Penginapan		√
5	Wisata Belanja	Garut	30	Oktober	Bus	Penginapan		√

Tabel 2. Hasil Rekomendasi Wisata

No	Jenis Wisata	Daerah Wisata	Calon Wisata	Waktu	Transportasi	Biaya Akomodasi	Baik	Buruk
1	Wisata Alam	Lembang	15	Agustus	1 Bus	3.000.000	√	
2	Petualangan	Ciwidey	5	Juni	Mobil	1.000.000	√	
3	Kuliner	Dago	10	Juli	Mobil	900.000	√	
4	Wisata Alam	Ciwidey	4	Juni	Motor	200.000		√
5	Wisata Alam	Garut	30	Oktober	Bus	4.500.000		√

## Kesimpulan

Kesimpulan yang dapat diperoleh dari penelitian ini berdasarkan pengujian adalah:

1. Berdasarkan pengujian Aplikasi *Travel Recommender* dapat memberikan rekomendasi wisata berdasarkan rencana wisata sebelumnya dengan tingkat keberhasilan 60% dan kegagalan 40%.
2. Berdasarkan pengujian *Case based reasoning* berhasil diimplementasikan dalam aplikasi *Travel Recommender*.

## Daftar Pustaka

- [1] Hayati, Nurul (2011) Metode Hybrid Nearest Neighbor Untuk Sistem Rekomendasi Pariwisata
- [2] Ramelo, Randy & Musrini Mira. (2011) > Kumpulan Tinjauan pustaka Indonesia Freelance.
- [3] Aamodt, A & Plaza, E, 1994, *Case Based Reasoning: Foundation Issues Methodological Variations, and System Approaches, AI Communication Vol 7 Nr, 1 March 1994, pp 39-59.*
- [4] Antonius, Rachmat. Sistem Bantu Diagnosis Penyakit Anjing 2012