

BAB II

LANDASAN TEORI DAN TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Penelitian tentang sistem pendukung keputusan metode *TOPSIS* ini memang bukan yang pertama kali dilakukan, sudah ada beberapa peneliti terdahulu terkait sistem seperti ini. Salah satunya adalah penelitian tentang Sistem Pendukung Keputusan Pemilihan Laptop Dengan Metode *TOPSIS* yang dilakukan oleh Kurniasih (2013). Penelitian ini bertujuan untuk membantu konsumen laptop memilih laptop dengan kriteria yang telah ditentukan dengan metode *TOPSIS*.

Penelitian lain yang berkaitan dengan penelitian ini adalah penelitian yang berjudul Penerapan Metode *TOPSIS* untuk Menentukan Prioritas Kondisi Rumah yang dilakukan oleh Chamid (2016). Penelitian ini bertujuan untuk membantu ibu rumah tangga menentukan prioritas mana yang harus didahulukan agar rumah sehat.

Selain itu ada penelitian tentang Pemilihan Supplier Suku Cadang Dengan Menggunakan Metode *TOPSIS* Pada PT. Delta Dunia Sandang Tekstil yang dilakukan Saraswati dan Adnan (2017). Penelitian ini bertujuan untuk membantu perusahaan PT. Delta Dunia Sandang Tekstil untuk memilih supplier yang cocok dan mampu mengimbangi proses produksi pada pabrik tersebut dengan metode *TOPSIS*.

Selanjutnya, ada Implementasi Metode *TOPSIS* Pada Sistem Pendukung Keputusan untuk Seleksi Siswa Kelas Unggulan yang dilakukan oleh Azizi (2014). Penelitian ini bertujuan untuk membantu salah satu MA (Madrasah Aliyah) di Banyuwangi dalam memilih siswa unggulan dengan metode *TOPSIS*.

Yang terakhir, ada penelitian tentang Sistem Pendukung Keputusan Pemilihan Loka-

si Wisata Daerah Istimewa Yogyakarta Dengan Metode *Analytical Hierarchy Procceess* (*AHP*) dan *Weighted Product* (*WP*) yang dilakukan oleh Firdausi (2017). Penelitian ini bertujuan untuk membuat sistem yang membantu para wisatawan yang datang ke Yogyakarta memilih objek wisata dengan dua metode, yaitu *AHP* dan *WP*. Sistem yang dihasilkan pada penelitian ini adalah sistem berbasis web dengan membandingkan hasil dari kedua metode tadi dengan parameter yang sama.

Sementara yang membedakan penelitian yang dilakukan kali ini dengan penelitian-penelitian sebelumnya adalah pada penelitian kali ini akan menerapkan metode *TOPSIS* dalam sistem pemilihan objek wisata di Daerah Istimewa Yogyakarta. Sistem yang dibuat akan berjalan pada sistem operasi berbasis *Android*. Tujuan penelitian ini adalah merancang dan mengimplementasi sistem untuk membantu wisatawan yang datang ke Daerah Istimewa Yogyakarta dalam memilih dan menentukan objek wisata mana yang akan dikunjungi. Secara ringkas, penelitian yang pernah dilakukan sebelumnya dan berkaitan dengan penelitian yang akan dibuat, disajikan pada Tabel 2.1 dan 2.2.

Tabel 2.1 Tinjauan Pustaka

No	Peneliti	Obyek Penelitian	Metode	Hasil
1	Kurniasih, 2013	Sistem Pendukung Keputusan Pemilihan Laptop	<i>TOPSIS</i>	Memperudah bagi pengguna yang ingin membeli komputer portabel (laptop) untuk memilih laptop sesuai dengan kebutuhan yang mereka inginkan.
2	Chamid, 2016	Prioritas Kondisi Rumah Sehat	<i>TOPSIS</i>	Menghasilkan sistem yang dapat menilai kondisi suatu rumah dan membantu ibu rumah tangga untuk menentukan prioritas mana yang harus didahulukan agar rumah sehat.
3	Saraswati dan Adnan, 2017	Pemilihan <i>Supplier</i> Suku Cadang pada PT. Delta Dunia Sandang Tekstil	<i>TOPSIS</i>	<i>Supplier</i> yang direkomendasikan untuk PT. Delta Dunia Sandang Tekstil adalah PT. Agape Trikarsa Libratama.

Tabel 2.2 Tinjauan Pustaka (Lanjutan)

No	Peneliti	Objek Penelitian	Metode	Hasil
4	Azizi, 2014	Seleksi Siswa Kelas Unggulan pada Salah Satu MA di Banyuwangi	<i>TOPSIS</i>	Sistem yang dibuat memudakan Madrasah Aliyah Al Amiriyah Darussalam Blongagung untuk memilih siswa masuk ke kelas unggulan.
5	Firdausi, 2017	Pemilihan Objek Wisata di Daerah Istimewa Yogyakarta	<i>Analytical Hierarchy Proccess (AHP)</i> dan <i>Weighted Product (WP)</i>	Berhasil merancang dan membangun sistem pendukung keputusan pemilihan lokasi wisata dengan obyek Provinsi Daerah Istimewa Yogyakarta menggunakan metode <i>AHP</i> dan <i>WP</i> .
6	Dwihastadi, 2019	Sistem Pendukung Keputusan Pemilihan Objek Wisata dengan berbasis <i>Android</i>	<i>TOPSIS</i>	Berhasil merancang dan membangun sistem pendukung keputusan pemilihan objek wisata di Daerah Istimewa Yogyakarta dengan metode <i>TOPSIS</i> yang berjalan pada sistem operasi <i>Android</i> .

Adapun yang membedakan antara penelitian yang diajukan dengan penelitian-penelitian sebelumnya adalah penelitian ini hanya menggunakan metode *TOPSIS* untuk menentukan rekomendasi lokasi wisata di Daerah Istimewa Yogyakarta sesuai kriteria yang diprioritaskan oleh wisatawan. Sistem yang dihasilkan dari penelitian ini berupa aplikasi *mobile* yang berjalan pada platform *android* sebagai pemandu bagi para wisatawan dalam menentukan lokasi wisatanya dan aplikasi web untuk admin jika nanti ingin menambah data lokasi wisata di Daerah Istimewa Yogyakarta. Penelitian dimulai dengan mengumpulkan data lokasi wisata di Daerah Istimewa Yogyakarta untuk menentukan kriteria lokasi wisata dan perhitungan menggunakan metode *TOPSIS*.

2.2. Landasan Teori

2.2.1. Sistem Pendukung Keputusan

1. Pengertian Sistem Pendukung Keputusan

Sistem Pendukung Keputusan (SPK) adalah sebuah sistem yang mampu memberikan kemampuan pemecahan masalah maupun kemampuan pengkomunikasian untuk masalah dengan kondisi semi terstruktur dan tak terstruktur. Sistem ini digunakan untuk membantu pengambilan keputusan dalam situasi semi terstruktur dan situasi yang tidak terstruktur, dimana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat (Turban, 2005).

Pada dasarnya sistem pendukung keputusan merupakan pengembangan lebih lanjut dari sistem manajemen terkomputerisasi yang dirancang sedemikian rupa sehingga bersifat interaktif dengan pemakainya. Sifat interaktif ini dimaksudkan untuk memudahkan integrasi antara berbagai komponen dalam proses pengambilan keputusan seperti prosedur, kebijakan, teknis, analisis, serta pengalaman

dan wawasan manajerial guna membentuk suatu kerangka keputusan yang bersifat fleksibel (Eniyanti, 2011).

Definisi mengenai sistem pendukung keputusan (SPK) yang ideal menurut Dwijaya (2010) yaitu :

- a SPK adalah sebuah sistem berbasis komputer dengan antarmuka antara mesin/komputer dan pengguna.
- b SPK ditujukan untuk membantu pembuat keputusan dalam menyelesaikan suatu masalah dalam berbagai level manajemen dan bukan untuk mengganti posisi manusia sebagai pembuat keputusan.
- c SPK mampu memberi alternatif solusi bagi masalah semi/tidak terstruktur baik bagi perseorangan atau kelompok dan dalam berbagai macam proses dan gaya pengambilan keputusan.
- d SPK menggunakan data, basis data dan analisa model-model keputusan.
- e SPK bersifat adaptif, efektif, interaktif, mudah digunakan dan fleksibel.
- f SPK menyediakan akses terhadap berbagai macam format dan tipe sumber data.

Tujuan dari pembuatan sistem pendukung keputusan yaitu (Turban, 2005):

- Membantu manajer membuat keputusan untuk memecahkan masalah yang sepenuhnya terstruktur dan tidak terstruktur.
- Mendukung penilaian manajer bukan mencoba menggantikannya. Sistem pendukung keputusan tidak dimaksudkan untuk menggantikan manajer. Komputer dapat diterapkan dalam menyelesaikan masalah yang terstruktur. Untuk masalah yang tidak terstruktur, manajer bertanggung jawab menerapkan penilaian, dan melakukan analisis. Komputer dan manajer bekerjasama sebagai

tim pemecahan masalah dalam memecahkan masalah yang berada di area semi terstruktur.

- Meningkatkan efektivitas pengambilan keputusan manajer dari pada efisiensinya. Tujuan utama sistem pendukung keputusan bukanlah proses pengambilan keputusan seefisien mungkin, tetapi seefektif mungkin.

Menurut Hermawan (2002) dalam Eniyanti (2011), proses pengambilan keputusan melalui beberapa tahap, antara lain :

a Tahap Penelusuran (*intelligence*)

Tahap ini pengambil keputusan mempelajari kenyataan yang terjadi, sehingga kita bisa mengidentifikasi masalah yang terjadi biasanya dilakukan analisis dari sistem ke subsistem pembentuknya sehingga didapatkan keluaran berupa dokumen pernyataan masalah.

b Tahap Desain

Dalam tahap ini pengambil keputusan menemukan, mengembangkan dan menganalisis semua pemecahan yang mungkin yaitu melalui pembuatan model yang bisa mewakili kondisi nyata masalah. Dari tahapan ini didapatkan keluaran berupa dokumen alternatif solusi.

c Tahap *Choice*

Dalam tahap ini pengambil keputusan memilih salah satu alternatif pemecahan yang dibuat pada tahap desain yang dipandang sebagai aksi yang paling tepat untuk mengatasi masalah yang sedang dihadapi. Dari tahap ini didapatkan dokumen solusi dan rencana implementasinya.

d Tahap Implementasi

Pengambil keputusan menjalankan rangkaian aksi pemecahan yang dipilih di tahap *choice*. Implementasi yang sukses ditandai dengan terjawabnya masalah yang dihadapi, sementara kegagalan ditandai masih adanya masalah yang sedang dicoba untuk diatasi, Dari tahap ini didapatkan laporan pelaksanaan solusi dan hasilnya.

2. Kelebihan dan Kekurangan Sistem Pendukung Keputusan

Secara global, dapat dikatakan bahwa tujuan dari Sistem Pendukung Keputusan adalah untuk meningkatkan kemampuan para pengambil keputusan dengan memberikan alternatif-alternatif keputusan yang lebih banyak atau lebih baik dan membantu untuk merumuskan masalah dan keadaan yang dihadapi. Dengan demikian Sistem Pendukung Keputusan dapat menghemat waktu, tenaga dan biaya. Jadi dapatlah dikatakan secara singkat bahwa tujuan Sistem Pendukung Keputusan adalah untuk meningkatkan efektivitas (*do the right things*) dan efisiensi (*do the things right*) dalam pengambilan keputusan. Walaupun demikian, penekanan dari suatu DSS adalah pada peningkatan efektivitas dari pengambilan keputusan dari pada efisiensinya.

(a) Kelebihan Sistem Pendukung Keputusan

Decision Support System (DSS) dapat memberikan beberapa keuntungan-keuntungan bagi pemakainya. Menurut Turban (1995: 87) maupun McLeod (1995: 103) keuntungan-keuntungan tersebut meliputi:

- Memperluas kemampuan pengambil keputusan dalam memproses data/informasi untuk pengambilan keputusan.
- Menghemat waktu yang dibutuhkan untuk memecahkan masalah, terutama berbagai masalah yang sangat kompleks dan tidak terstruktur.

- Menghasilkan solusi dengan lebih cepat dan hasilnya dapat diandalkan.
- Mampu memberikan berbagai alternatif dalam pengambilan keputusan, meskipun seandainya DSS tidak mampu memecahkan masalah yang dihadapi oleh pengambil keputusan, namun dapat digunakan sebagai stimulan dalam memahami persoalan.
- Memperkuat keyakinan pengambil keputusan terhadap keputusan yang diambilnya.
- Memberikan keuntungan kompetitif bagi organisasi secara keseluruhan dengan penghematan waktu, tenaga dan biaya.

(b) Kekurangan Sistem Pendukung Keputusan

Walaupun dirancang dengan sangat teliti dan mempertimbangkan seluruh faktor yang ada, menurut Turban (1995: 250) DSS mempunyai kelemahan atau keterbatasan, diantaranya yaitu:

- Ada beberapa kemampuan manajemen dan bakat manusia yang tidak dapat dimodelkan, sehingga model yang ada dalam sistem tidak semuanya mencerminkan persoalan sebenarnya.
- DSS terbatas untuk memberikan alternatif dari pengetahuan yang diberikan kepadanya (pengetahuan dasar serta model dasar) pada waktu perancangan program tersebut.
- Proses-proses yang dapat dilakukan oleh DSS biasanya tergantung juga pada kemampuan perangkat lunak yang digunakan
- Harus selalu diadakan perubahan secara kontinyu untuk menyesuaikan dengan keadaan lingkungan yang terus berubah agar sistem tersebut *up to date*.

- Bagaimanapun juga harus diingat bahwa DSS dirancang untuk membantu/mendukung pengambilan keputusan dengan mengolah informasi dan data yang diperlukan, dan bukan untuk mengambil alih pengambilan keputusan.

3. Penentuan Kriteria Sistem Pendukung Keputusan

Sifat-sifat yang harus diperhatikan dalam memilih kriteria dalam pengambilan keputusan menurut (Supriadi, 2010) adalah sebagai berikut:

(a) Lengkap

Kriteria yang dipilih dapat mencakup seluruh aspek penting dalam persoalan tersebut. Suatu set kriteria disebut lengkap apabila set ini dapat menunjukkan seberapa jauh seluruh tujuan dapat dicapai.

(b) Operasional

Kriteria yang baik harus dapat digunakan dalam analisis. Sifat operasional ini mencakup beberapa pengertian, antara lain bahwa set kriteria ini harus mempunyai arti bagi pengambilan keputusan, sehingga ia dapat benar-benar mempengaruhi nilai terhadap alternatif yang ada. Selain itu, jika tujuan pengambilan keputusan ini harus dapat digunakan sebagai saran untuk meyakinkan pihak lain, maka set kriteria ini harus dapat digunakan sebagai sarana untuk memberikan penjelasan atau untuk berkomunikasi.

(c) Tidak Berlebihan

Kriteria yang dipilih tidak berlebihan untuk menghindari perhitungan yang berulang. Proses menentukan set kriteria diusahakan menghindari kriteria yang mengandung pengertian yang sama.

(d) Minimum

Jumlah kriteria harus minimum dengan tujuan agar lebih mengkomprehensifkan persoalan. Semakin banyak kriteria yang dilibatkan maka semakin sukar pula untuk dapat menghayati permasalahan dengan baik, lebih jauh lagi, jumlah perhitungan yang diperlukan dalam analisis akan semakin banyak.

2.2.2. *Multiple Attribute Decision Making (MADM)*

Multiple Attribute Decision Making (MADM) adalah metode pengambilan keputusan dengan mengambil kriteria yang dinilai secara subyektif sesuai masalah yang akan diselesaikan, tidak memerlukan analisis matematis yang terlalu banyak dan digunakan untuk pemilihan alternatif dalam jumlah sedikit. Beberapa teknik dari *Multiple Attribute Decision Making (MADM)* seperti berikut:

a. Metode *Simple Additive Weighting (SAW)*

- Sering disebut dengan istilah metode penjumlahan terbobot.
- Konsep dasarnya adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut.
- Metode ini membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada.

b. *Weighted Product (WP)*

- Metode ini menggunakan perkalian untuk menghubungkan rating atribut, dimana rating setiap atribut harus dipangkatkan dulu dengan bobot atribut yang bersangkutan.
- Proses ini hanya dengan proses normalisasi.

c. *ELECTRE*

- Merupakan salah satu metode pengambilan keputusan multikriteria berdasarkan pada konsep outranking dengan menggunakan perbandingan berpasangan dari alternatif-alternatif berdasarkan setiap kriteria yang sesuai.
- Digunakan untuk kasus-kasus dengan banyak alternatif namun hanya sedikit kriteria yang dilibatkan.

d. *Technique for Order Preference by Similarity of Ideal Solution (TOPSIS)*

- Salah satu metode pengambilan keputusan multikriteria yang pertama kali diperkenalkan oleh Yoon dan Hwang (1981).
- Menggunakan prinsip bahwa alternatif yang terpilih harus mempunyai jarak terdekat dari solusi ideal positif dan terjauh dari solusi ideal negatif dari sudut pandang geometris dengan menggunakan jarak Euclidean untuk menentukan kedekatan relatif dari suatu alternatif dengan solusi optimal.
- Metode ini banyak digunakan untuk menyelesaikan pengambilan keputusan secara praktis, karena konsepnya sederhana dan mudah dipahami, komputasinya efisien, dan memiliki kemampuan mengukur kinerja relatif dari alternatif-alternatif keputusan.

e. *Analytical Hierarchy Process (AHP)*

- Merupakan suatu model pendukung keputusan yang dikembangkan oleh Thomas L. Saaty.
- menguraikan masalah multifaktor atau multi kriteria yang kompleks menjadi suatu hirarki.

2.2.3. Metode *Topsis* (*Technique For Others Reference by Similarity to Ideal Solution*)

Adalah salah satu metode pengambilan keputusan multikriteria yang pertama kali diperkenalkan oleh Yoon dan Hwang pada tahun 1981. Metode ini merupakan salah satu metode yang banyak digunakan untuk menyelesaikan pengambilan keputusan secara praktis. *TOPSIS* memiliki konsep dimana alternatif yang terpilih merupakan alternatif terbaik yang memiliki jarak terpendek dari solusi ideal positif dan jarak terjauh dari solusi ideal negatif. Semakin banyaknya faktor yang harus dipertimbangkan dalam proses pengambilan keputusan, maka semakin relatif sulit juga untuk mengambil keputusan terhadap suatu permasalahan. Apalagi jika upaya pengambilan keputusan dari suatu permasalahan tertentu, selain mempertimbangkan berbagai faktor/kriteria yang beragam, juga melibatkan beberapa orang pengambil keputusan. Permasalahan yang demikian dikenal dengan permasalahan *multiple criteria decision making (MCDM)*. Dengan kata lain, MCDM juga dapat disebut sebagai suatu pengambilan keputusan untuk memilih alternatif terbaik dari sejumlah alternatif berdasarkan beberapa kriteria tertentu. Metode *TOPSIS* digunakan sebagai suatu upaya untuk menyelesaikan permasalahan *multiple criteria decision making*. Hal ini disebabkan konsepnya sederhana dan mudah dipahami, komputasinya efisien dan memiliki kemampuan untuk mengukur kinerja relatif dari alternatif-alternatif keputusan.

Prosedur dan Langkah Penyelesaian Metode *TOPSIS*

a. Membangun *normalized decision matrix*

Elemen r_{ij} hasil dari normalisasi *decision matrix* R dengan metode *Euclidean length*

of a vector dijelaskan pada Rumus 2.1.

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}} \quad (2.1)$$

b. Membangun *weighted normalized decision matrix*

Matriks ternormalisasi terbobot didapat dengan mengalikan matriks ternormalisasi dengan bobot yang dimasukkan. Rumus untuk mendapat y_{ij} dijelaskan dalam Rumus 2.2.

$$y_{ij} = w_i r_{ij} \quad (2.2)$$

c. Menentukan matriks solusi ideal dan matriks solusi ideal negatif

Solusi ideal positif (A^+) didapatkan berdasarkan Rumus 2.3.

$$A^+ = (y1^+, y2^+, y3^+, \dots, yn^+) \quad (2.3)$$

Nilai solusi ideal positif adalah nilai *max* kriteria dari seluruh alternatif jika kriteria bersifat *benefit* dan *min* kriteria jika kriteria bersifat *cost*. Solusi ideal negatif (A^-) didapatkan berdasarkan Rumus 2.4.

$$A^- = (y1^-, y2^-, y3^-, \dots, yn^-) \quad (2.4)$$

Nilai solusi ideal negatif adalah nilai *max* kriteria dari seluruh alternatif jika kriteria bersifat *cost* dan *min* kriteria jika kriteria bersifat *benefit*.

d. Menentukan jarak antara nilai setiap alternatif dengan matriks solusi ideal positif dan matrik ideal negatif

Jarak antara alternatif dengan solusi ideal positif dijelaskan pada Rumus 2.5.

$$D_i^+ = \sqrt{\sum_{i=1}^n (y_i^+ - y_{ij})^2}, i = 1, 2, 3, \dots, m \quad (2.5)$$

Jarak antara alternatif dengan solusi ideal negatif dijelaskan pada Rumus 2.6.

$$D_i^- = \sqrt{\sum_{j=1}^n (y_{ij} - y_i^-)}, i = 1, 2, 3, \dots, m \quad (2.6)$$

e. Menentukan nilai preferensi untuk setiap alternatif

Kedekatan setiap alternatif terhadap solusi ideal dihitung berdasarkan Rumus 2.7.

$$V = \frac{D_i^-}{D_i^+ + D_i^-}, i = 1, 2, 3, \dots, m \quad (2.7)$$

Nilai preferensi untuk setiap alternatif merupakan hasil akhir dari perhitungan metode *TOPSIS*, semakin tinggi nilai nya maka alternatif tersebut merupakan alternatif yang diinginkan.

2.2.4. UML(*Unified Modeling Language*)

UML adalah bahasa grafis untuk mendokumentasi, menspesifikasikan, dan membangun sistem perangkat lunak. UML berorientasi objek, menerapkan banyak level abstraksi, tidak bergantung proses pengembangan, tidak bergantung bahasa dan teknologi, pemanduan beberapa notasi di beragam metodologi, usaha bersama dari banyak pihak, didukung oleh kakas-kakas yang diintergrasikan ewat XML (XMI). Standar UML dikelola oleh OMG Object Management Group) (Hariyanto, 2004).

1. *Class Diagram*

Diagram ini menunjukkan sekumpulan kelas, interface, dan kolaborasi dan keterhubungannya. Diagram kelas ditujukan untuk pandangan statik terhadap sistem. Notasi simbol yang terdapat pada diagram kelas disajikan dalam Gambar 2.1.

2. *Use Case Diagram*

Diagram ini menunjukkan sekumpulan kasus fungsional dan aktor (jenis kelas khusus) dan keterhubungannya. Notasi simbol yang terdapat pada diagram *usecase* disajikan dalam Gambar 2.2.

3. *Sequence Diagram*








Diagram ini menunjukkan interaksi yang terjadi antar objek. Diagram ini merupakan pandangan dinamis terhadap sistem. Diagram ini menekankan pada basis keberurutan waktu dari pesan-pesan yang terjadi. Notasi simbol yang terdapat pada diagram *sequence* disajikan dalam Gambar 2.3.

4. *Activity Diagram*






Diagram ini untuk menunjukkan aliran aktivitas di sistem. Diagram ini adalah pandangan dinamis terhadap sistem. Diagram ini penting untuk memodelkan fungsi sistem dan menekankan pada aliran kendali antara objek-objek. Notasi simbol yang terdapat pada diagram *activity* disajikan dalam Gambar 2.4.

2.2.5. ERD (Entity Relationship Diagram)











Entity Relationship Diagram (ERD) merupakan teknik yang digunakan untuk memodelkan aliran data dari suatu organisasi. Teknik peraga ini memeberikan gambaran akan perancangan database suatu sistem informasi yang akan dikembangkan. ERD dengan model pendukungnya bersama-sama digunakan sebagai spesifikasi untuk suatu database. (Brandy dan Loonam, 2010). Kmponen yang ada dalam ERD, yaitu entitas, relasi dan atribut yang disajikan dalam Gambar 2.5.

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya






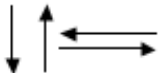
Gambar 2.1 Simbol-simbol dalam *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
3		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
4		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
5		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor

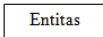
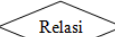
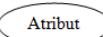
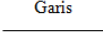
Gambar 2.2 Simbol-simbol dalam *Use Case Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>Sistem</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (<i>sinergi</i>).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Gambar 2.3 Simbol-simbol dalam *Sequence Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		Activity	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		Action	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		Initial Node	Bagaimana objek dibentuk atau diawali.
4		Activity Final Node	Bagaimana objek dibentuk dan diakhiri
5		Decision	Digunakan untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu
6		Line Connector	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya

Gambar 2.4 Simbol-simbol dalam *Activity Diagram*

Notasi	Keterangan
	Entitas adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai.
	Relasi menunjukkan adanya hubungan di antara sejumlah entitas yang berbeda.
	Atribut berfungsi mendeskripsikan karakter entitas (atribut yang berfungsi sebagai <i>key</i> diberi garis bawah).
	Garis sebagai penghubung antara relasi dan entitas atau relasi dan entitas dengan atribut.

Gambar 2.5 Komponen dan Penjelasan ERD

2.2.6. PHP (Hypertext Preprocessor)

Singkatan dari Hypertext Preprocessor yang digunakan sebagai bahasa script serverside dalam pengembangan Web yang disisipkan pada dokumen HTML. Berbeda dengan HTML yang hanya bisa menampilkan konten statis, PHP bisa berinteraksi dengan database, file dan folder, sehingga membuat PHP bisa menampilkan konten yang dinamis dari sebuah website. Pada dasarnya PHP merupakan bahasa pemrograman disisi server (daqiqil,2011). PHP digunakan untuk memanggil database yang ada dalam pihak server dengan menggunakan script khusus yang mirip dengan C++. Dalam pemanggilan data, PHP tidak berdiri sendiri dan harus

didiukung oleh aplikasi lain, misalnya HTML (Hyper Text Markup Language) sebagai pengaturan interface serta MySQL sebagai aplikasi database yang terintegrasi dengan PHP. PHP mendukung banyak paket basis data yang berlisensi maupun open source seperti PostgreSQL, mSQL, oracle, MySQL, Informic, Microsoft SQL Server, dan lain-lain. PHP mendukung aplikasi web database skala besar.

2.2.7. Basis Data dan DBMS (Data Base Management System)

Secara sederhana database atau basis data dapat diungkapkan sebagai suatu pengorganisasian data dengan bantuan komputer yang memungkinkan data dapat diakses dengan mudah dan cepat. Menurut date, sistem basis data pada dasarnya adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara informasi dan membuat informasi tersebut tersedia saat dibutuhkan (Kadir, 2002).

Database diimplementasikan dalam sebuah perangkat lunak untuk manajemen database tersebut, perangkat lunak yang digunakan untuk manajemen database disebut DBMS (Database Manajemen System). DBMS adalah perangkat lunak yang di desain untuk mengelola database. Secara lebih rinci DBMS merupakan kumpulan software program yang sangat kompleks untuk mengontrol kumpulan data dan sebagai alat penyimpanan data di dalam database.

2.2.8. MySQL

MYSQL merupakan salah satu DBMS yang sangat populer dikalangan pemrograman web yang bersifat open source. MYSQL awalnya dibuat oleh perusahaan konsultan bernama TEX yang berlokasi di Swedia. Saat ini pengembangan MySQL berada dibawah MySQLAB. MySQL menggunakan SQL (Struktur Query Language) Sebagai bahasa pemrograman untuk mengakses data dari server MySQL.

2.2.9. Kotlin

Kotlin adalah sebuah bahasa pemrograman dengan pengetikan statis yang berjalan pada Mesin Virtual Java ataupun menggunakan kompiler LLVM yang dapat pula dikompilasikan kedalam bentuk kode sumber JavaScript. Pengembang utamanya berasal dari tim programmer dari JetBrains yang bermarkas di Rusia. Meskipun sintaksisnya tidak kompatibel dengan bahasa Java, Kotlin didesain untuk dapat bekerja sama dengan kode bahasa Java dan bergantung kepada kode bahasa Java dari Kelas Pustaka Java yang ada, seperti berbagai framework Java yang ada. Tim Pengembang memutuskan menamakannya Kotlin dengan mengambil nama dari sebuah pulau di Rusia, sebagaimana Java yang mengambil nama dari pulau Jawa di Indonesia. Setelah Google mengumumkan bahwa Kotlin menjadi bahasa kelas satu bagi Android, maka bersama Java dan C++, Kotlin menjadi bahasa resmi untuk pengembangan aplikasi-aplikasi Android.

2.2.10. Android

Android merupakan sistem operasi yang memang khusus dirancang untuk smartphone dan tablet. Sistem Android ini memiliki basis Linux yang mana dijadikan sebagai pondasi dasar dari sistem operasi Android. Linux sendiri merupakan sistem operasi yang memang khusus dirancang untuk komputer. Android memang dirancang untuk dipasang pada perangkat-perangkat mobile touchscreen (smartphone dan tablet). Sehingga sistem operasi yang berada di dalam smartphone saat ini memang menyesuaikan dari spesifikasi kelas low-end hingga high-end. Sehingga perkembangan sistem android memang cukup meningkat tajam. Android merupakan sistem operasi yang terbuka (open source) yang mana berarti jika pihak Google memperbolehkan dan membebaskan bagi pihak manapun untuk dapat mengembangkan sistem operasi tersebut. Bahkan anda sendiri pun juga dapat mengembangkan sistem android yang memang sesuai dengan keinginan anda.

2.2.11. API (Application Programming Interface)

API adalah singkatan dari *Application Programming Interface*, dan memungkinkan *developer* untuk mengintegrasikan dua bagian dari aplikasi atau dengan aplikasi yang berbeda secara bersamaan. *API* terdiri dari berbagai elemen seperti *function*, *protocols*, dan *tools* lainnya yang memungkinkan *developers* untuk membuat aplikasi. Tujuan penggunaan *API* adalah untuk mempercepat proses *development* dengan menyediakan *function* secara terpisah sehingga *developer* tidak perlu membuat fitur yang serupa. Penerapan *API* akan sangat terasa jika fitur yang diinginkan sudah sangat kompleks, tentu membutuhkan waktu untuk membuat yang serupa dengannya.

2.2.12. Extreme Programming (XP)

Metode XP pertama kali ditemukan oleh Kent Beck, seorang pakar software engineering yang bekerja dengan Chrysler yang membuat proyek C3 (Chrysler Comprehensive Compensation). Pada saat itu proyek Chrysler terancam gagal sebelum Kent Beck di kontrak, kemudian saat Kent Beck memegang proyek ini, Kent Beck bersama Ron Jeffries menyelesaikan proyek sesuai dengan target dengan menggunakan berbagai metode. Kemudian kumpulan metode ini disebut dengan metode XP. Kent Beck membuat perubahan pada pembangunan proyek dengan membuatnya lebih efisien, adaptif, dan fleksibel. XP memiliki 4 nilai dasar yang menjadi inti pokok metode XP [4] yaitu : Communication (Komunikasi), Simplicity (Kesederhanaan), Feedback (Umpan Balik), dan Courage (Keberanian). Keempat nilai dasar ini menunjukkan bahwa XP bersifat fleksibel terhadap perubahan-perubahan yang diminta oleh klien.

Adapun tahapan dalam pembangunan aplikasi SPK dengan XP adalah sebagai berikut :

1. **Tahap *Planning***

Tahapan ini adalah tahapan dimana klien menuliskan kebutuhan-kebutuhan dari sistem yang paling mendasar. Setiap kebutuhan yang dituliskan oleh klien akan dibuat dalam bentuk modul yang sederhana atau disebut juga dengan User Stories. Hasil dari tahapan eksplorasi adalah mengetahui dokumentasi atas visi dan ruang lingkup pekerjaan.

2. **Tahap *Design***

Tahapan ini akan memperkirakan kebutuhan bisnis, kebutuhan *user*, dan kebutuhan sistem. Tahapan ini juga akan menghasilkan penjadwalan yang menggambarkan perencanaan waktu pelaksanaan pembangunan sistem.

3. **Tahap *Coding***

Pada tahapan ini, akan terjadi beberapa kali iterasi, setiap iterasi terdiri dari 3 tahapan yaitu analisis sistem, desain sistem, dan pembuatan dan pengujian sistem.

4. **Tahap *Testing***

Tahapan produksi akhir adalah tahapan dimana sistem sudah siap untuk di release. Tahapan ini akan melakukan testing terhadap keseluruhan sistem yang telah dibuat kepada klien.

BAB III

METODE PENGEMBANGAN SISTEM

3.1. Studi Pendahuluan

Pengembangan sistem ini dilaksanakan untuk menghasilkan rekomendasi lokasi wisata di Provinsi Daerah Istimewa Yogyakarta yang diperoleh dari proses pengambilan keputusan pemilihan lokasi wisata dengan menentukan kriteria yang menjadi prioritas. Dalam pengembangan sistem ini penulis menggunakan metode TOPSIS dalam menentukan rekomendasi objek wisata yang berjalan dalam platform smartphone berbasis android.

Dalam sistem yang dibangun penulis menggunakan lima kriteria *benefit* dan dua kriteria *cost*. Dua kriteria *cost* yaitu jarak objek wisata dari user dan HTM (Harga Tiket Masuk). Sementara lima kriteria *benefit* tersebut adalah, transportasi, fasilitas, keamanan, keindahan dan kebersihan. Penulis merancang sistem dengan memberikan akses langsung terhadap setiap user untuk menentukan seberapa pentingkah ketujuh kriteria tadi saat akan menentukan objek wisata yang diinginkan. Kriteria-kriteria dasar yang disebutkan didapatkan oleh penulis dengan menggunakan kuisioner. Kriteria-kriteria tersebut disajikan dalam Tabel.3.1.

Tabel 3.1 Tabel Kriteria yang Digunakan

Kriteria	Keterangan
Keindahan	<i>Benefit</i>
Keamanan	<i>Benefit</i>
Kebersihan	<i>Benefit</i>
Fasilitas	<i>Benefit</i>

Tabel 3.2 Tabel Kriteria yang Digunakan (Lanjutan)

Kriteria	Keterangan
Transportasi	<i>Benefit</i>
HTM	<i>Cost</i>
Jarak	<i>Cost</i>

3.2. Tahap Pengumpulan Data

Tahap pengumpulan data dilakukan oleh peneliti melalui empat cara, yaitu dengan melakukan studi literatur atau kepustakaan, wawancara, kuisisioner dan juga observasi langsung terhadap objek penelitian.

3.2.1. Studi Literatur

Metode pengumpulan data yang dilakukan melalui membaca dan mempelajari referensi-referensi berupa jurnal ilmiah dan skripsi. Fasilitas internet yang dimanfaatkan untuk media sebagai pencari data objek wisata atau informasi referensi yang dipublikasikan di dunia maya yang berkaitan dengan obyek penelitian.

3.2.2. Metode Wawancara

Wawancara yang dilakukan melalui tanya jawab dengan pihak yang terkait untuk memperoleh data-data yang dibutuhkan. Wawancara ini dilakukan dengan beberapa wisatawan yang bertindak sebagai user.

3.2.3. Kuisisioner

Kuisisioner dilakukan melalui pemberian daftar pertanyaan terhadap responden yang terkait untuk memperoleh data-data yang dibutuhkan. Kuisisioner dilakukan dengan meng-

gunakan Google Form, dengan 115 respon.

3.2.4. Observasi

Metode pengumpulan data yang dilakukan dengan melakukan pengamatan terhadap objek penelitian.

3.3. Metode Pengembangan Sistem

Metode pengembangan sistem ini menggunakan metode *Extreme Programming*. Metode ini digunakan karena cocok dengan sistem yang akan dibangun karena menggunakan kerangka yang runtut, adaptif dan proses yang lebih cepat serta fleksibel dibandingkan dengan metode lain. Sehingga penulis menganggap metode ini sangat baik apabila diterapkan pada sistem skala kecil yang diharapkan minim risiko.

Seperti dalam penjelasan di landasan teori tentang metode *Extreme Programming* sebelumnya, tahapan-tahapan yang akan dilakukan pada penelitian ini adalah 4 (empat) tahap, yaitu:

1. Tahap *Planning*

Fase ini dilakukan dengan mengumpulkan kebutuhan sistem secara lengkap kemudian dianalisis dan dapat didefinisikan kebutuhan yang harus dipenuhi oleh sistem yang akan dibangun. Tahap ini penulis melakukan observasi tentang sistem yang sudah berjalan. Lalu menulis dan merencanakan kebutuhan-kebutuhan apa saja yang sekiranya akan dibutuhkan *user* pada sistem yang akan dibuat.

2. Tahap *Design*

Setelah mendapatkan apa saja kebutuhan *user* yang ingin didapatkan dari sistem, selanjutnya penulis melakukan perencanaan dan desain sistem secara keseluruhan.

an. Mulai dari rancangan *database*, *UML*, tampilan antarmuka dan rancangan-rancangan lainnya yang perlu dibuat.

3. Tahap *Coding*

Dalam tahap ini, akan dilakukan pengimplementasian sistem yang telah dirancang sebelumnya. Pada sistem kali ini, sistem akan dibuat dengan menggunakan bahasa pemrograman *Kotlin*, *PHP* dengan *framework Laravel* dan menggunakan *database MySQL*.

4. Tahap *Testing*

Setelah sistem siap dibangun, maka tahap selanjutnya adalah melakukan pengujian kepada *user*. Jika *user* memiliki masukan maka tahap-tahap sebelumnya akan diulangi dari awal sampai tercapainya sistem yang diinginkan sesuai dengan penelitian ini.

3.4. Kebutuhan Pengembangan Sistem

3.4.1. Perangkat Keras (*Hardware*)

Adapun perangkat keras (*hardware*) yang digunakan dalam perancangan sistem ini adalah :

1. Laptop Lenovo Thinkpad E455
2. RAM 8 GB
3. Processor AMD A-10 7300 Quad-Core 1.9 ghz
4. SSD Samsung 850 EVO 512 GB
5. Asus Zenfone Max Pro M1

3.4.2. Perangkat Lunak(*Software*)

Adapun perangkat lunak (*software*) yang digunakan dalam perancangan sistem ini adalah :

1. Sistem Operasi Linux Deepin 15.9
2. Bahasa Pemrograman PHP 7.2
3. Web Server Apache
4. XAMPP
5. Database MySQL
6. Visual Studio Code
7. Google Chrome Web Browser
8. Android Studio 3.3
9. Bahasa Pemrograman Kotlin

BAB IV

ANALISIS DAN PERANCANGAN SISTEM

4.1. Analisis Kebutuhan Sistem

Analisis kebutuhan dan perancangan sistem adalah hal pertama yang harus dilakukan sebelum melakukan implementasi pengembangan sistem lebih lanjut. Analisis kebutuhan sistem bertujuan untuk mengidentifikasi apa saja yang dibutuhkan oleh sistem yang akan di buat, dan jika ada maka juga akan dilakukan identifikasi tentang kekurangan sistem yang telah berjalan. Sehingga sistem yang nantinya dibuat akan benar-benar memecahkan masalah dan menyempurnakan sistem yang telah ada sebelumnya.

4.1.1. Analisis Sistem yang Berjalan

Dinas Pariwisata Yogyakarta telah memiliki situs resmi tentang wisata-wisata yang ada di Daerah Istimewa Yogyakarta yang di kelola langsung oleh Dinas Pariwisata Yogyakarta. Dalam situs tersebut, telah mencakup beberapa hal berikut :

1. Sudah tersedia data-data wisata di Daerah Istimewa Yogyakarta yang terbagi menjadi 5 kategori dan juga ada kategori wisata kuliner yang terpisah dari lainnya.
2. Sistem juga menyediakan data-data akomodasi tempat menginap di Daerah Istimewa Yogyakarta.
3. Terdapat juga peta wisata di Daerah Istimewa Yogyakarta dan Info Event yang menarik.
4. Dinas Pariwisata juga menyediakan berita-berita seputar wisata di website nya.

4.1.2. Analisis Masalah

1. Data objek wisata yang disajikan di Web Dinas Pariwisata Daerah kurang lengkap.
2. Web Dinas Pariwisata Dinas Daerah Istimewa Yogyakarta belum memiliki Aplikasi Android untuk menyediakan data-data wisata bagi para wisatawan yang mengunjungi Yogyakarta.
3. Web Dinas Pariwisata Yogyakarta belum memiliki sistem yang dapat merekomendasikan objek wisata kepada user.

4.1.3. Analisis Pengguna Sistem

Sistem pendukung keputusan yang akan dibangun hanya memiliki dua jenis pengguna dengan hak akses yang berbeda. Kedua pengguna tersebut adalah :

1. Admin

Admin disini bertindak sebagai pengguna yang memiliki hak akses untuk memasukkan data-data objek wisata yang nantinya akan dilihat oleh *user*.

2. User

User disini adalah pengguna yang menggunakan aplikasi mobile sistem ini. *User* dapat menginputkan kriteria yang diinginkan dan memberi penilaian terhadap 3 aspek (keindahan, kebersihan dan keamanan) objek wisata.

4.1.4. Analisis Sistem Usulan

Berdasarkan analisis yang telah dilakukan diatas, maka sistem yang diusulkan adalah sebagai berikut :

1. Sistem dapat menyimpan data objek wisata secara detail.

2. Sistem dapat menampilkan data objek wisata kepada *user* secara menarik.
3. Sistem dapat mengolah data sehingga mampu memberikan rekomendasi objek wisata.
4. Perhitungan TOPSIS di lakukan pada aplikasi mobile sehingga lebih cepat dan tidak memberatkan kinerja server.
5. Sistem dapat memberikan rekomendasi objek wisata sesuai dengan kriteria yang dimasukkan oleh *user* secara cepat dan akurat.

4.1.5. Analisis Kebutuhan Sistem

Untuk mempermudah menganalisis sebuah sistem dibutuhkan dua jenis kebutuhan. Kebutuhan fungsional dan kebutuhan nonfungsional. Kebutuhan fungsional adalah kebutuhan yang berisi proses-proses atau fitur apa saja yang nantinya dilakukan oleh sistem. Sedangkan kebutuhan nonfungsional adalah kebutuhan yang menitikberatkan pada properti perilaku yang dimiliki oleh sistem.

1. Kebutuhan Fungsional

- a Sistem dapat menyimpan data objek wisata secara detail.
- b *User* dapat melihat data objek wisata yang disimpan.
- c *User* dapat menambahkan penilaian terhadap 3 aspek (keindahan, keamanan dan kebersihan) pada objek wisata.
- d Sistem dapat memberikan rekomendasi objek wisata berdasarkan hasil penghitungan metode TOPSIS terhadap objek wisata.

2. Kebutuhan Non-Fungsional

- a Operasional, aplikasi mobile dalam sistem dapat digunakan dimana saja selama ada koneksi internet.
- b Praktis, karena aplikasi untuk *end-user* dibuat dalam bentuk mobile aplikasi harus mudah digunakan.
- c Aplikasi mobile harus memiliki design interface yang menarik sehingga wisatawan berminat saat melihat objek wisata dalam aplikasi.
- d *Sharing*, aplikasi dapat membagi pengalaman wisatawan/user kepada wisatawan lain yang ingin mengunjungi objek wisata.

4.1.6. Analisis Kriteria yang Digunakan

Dari keseluruhan analisis yang dilakukan oleh peneliti diatas, maka peneliti menyimpulkan untuk melakukan pengumpulan data menggunakan kuisisioner, dan kuisisioner tersebut menghasilkan beberapa kriteria yaitu jarak, HTM (harga tiket masuk), kemudahan transportasi, fasilitas, keindahan, kebersihan dan keamanan. Untuk penjelasan dari kriteria-kriteria tersebut sebagai berikut :

1. Kriteria Jarak

Kriteria jarak adalah jarak pengguna / *end-user* terhadap suatu objek wisata. Semakin jauh jarak pengguna terhadap objek wisata, semakin kecil nilai kriteria ini. Nilai kriteria jarak disajikan dalam Tabel 4.1.

Tabel 4.1 Tabel Kriteria Jarak

Kriteria	Tipe	Keterangan Nilai	Nilai
Jarak	Cost	kurang dari 18 km	1
		19-36 km	2
		37-54 km	3
		55-72 km	4
		lebih dari 73 km	5

2. Kriteria HTM (Harga Tiket Masuk)

Kriteria htm adalah kriteria harga tiket masuk sebuah objek wisata. Semakin mahal htm maka semakin kecil nilai kriteria nya. Nilai kriteria htm disajikan dalam Tabel 4.2.

Tabel 4.2 Tabel Kriteria HTM

Kriteria	Tipe	Keterangan Nilai	Nilai
HTM	Cost	kurang dari Rp. 20.000,00	1
		Rp.20.000-40.000	2
		Rp.40.000-60.000	3
		Rp.60.000-80.000	4
		lebih dari Rp.80.000	5

3. Kriteria Keamanan

Kriteria keamanan adalah kriteria seberapa aman sebuah objek wisata. Keamanan dapat dilihat dari aspek keamanan tempat parkir, keamanan saat menggunakan wahana dll. Nilai kriteria ini ditentukan oleh *user* yang menilai dari 1 sampai 5 bagaimana keamanan di sebuah objek wisata. Semakin aman sebuah objek wisata maka semakin besar nilai kriteria nya. Nilai kriteria keamanan disajikan dalam tabel 4.3.

Tabel 4.3 Tabel Kriteria Keamanan

Kriteria	Tipe	Nilai
Keamanan	Benefit	1 - 5

4. Kriteria Keindahan

Kriteria keindahan adalah kriteria seberapa indah sebuah objek wisata. Nilai kriteria ini ditentukan oleh *user* yang menilai dari 1 sampai 5 bagaimana keindahan di sebuah objek wisata. Semakin indah sebuah objek wisata maka semakin besar nilai kriteria nya. Nilai kriteria keindahan disajikan dalam Tabel 4.4.

Tabel 4.4 Tabel Kriteria Keindahan

Kriteria	Tipe	Nilai
Keindahan	Benefit	1 - 5

5. Kriteria Kebersihan

Kriteria kebersihan adalah kriteria seberapa bersih sebuah objek wisata. Nilai kriteria ini ditentukan oleh *user* yang menilai dari 1 sampai 5 bagaimana kebersihan di sebuah objek wisata. Semakin bersih sebuah objek wisata maka semakin besar nilai kriteria nya. Nilai kriteria kebersihan disajikan dalam tabel 4.5.

Tabel 4.5 Tabel Kriteria Kebersihan

Kriteria	Tipe	Nilai
Kebersihan	Benefit	1 - 5

6. Kriteria Fasilitas

Kriteria fasilitas adalah kriteria seberapa lengkap fasilitas sebuah objek wisata. Fasilitas yang di maksud dalam kriteria ini yaitu mushola, tempat parkir, toilet, restoran dan penginapan. Semakin lengkap fasilitas sebuah objek wisata maka semakin besar nilai kriteria nya. Nilai kriteria fasilitas disajikan dalam Tabel 4.6.

Tabel 4.6 Tabel Kriteria Fasilitas

Kriteria	Tipe	Nilai
Fasilitas	Benefit	1 - 5

7. Kriteria Transportasi

Kriteria transportasi adalah kriteria seberapa mudah sebuah objek wisata dijangkau

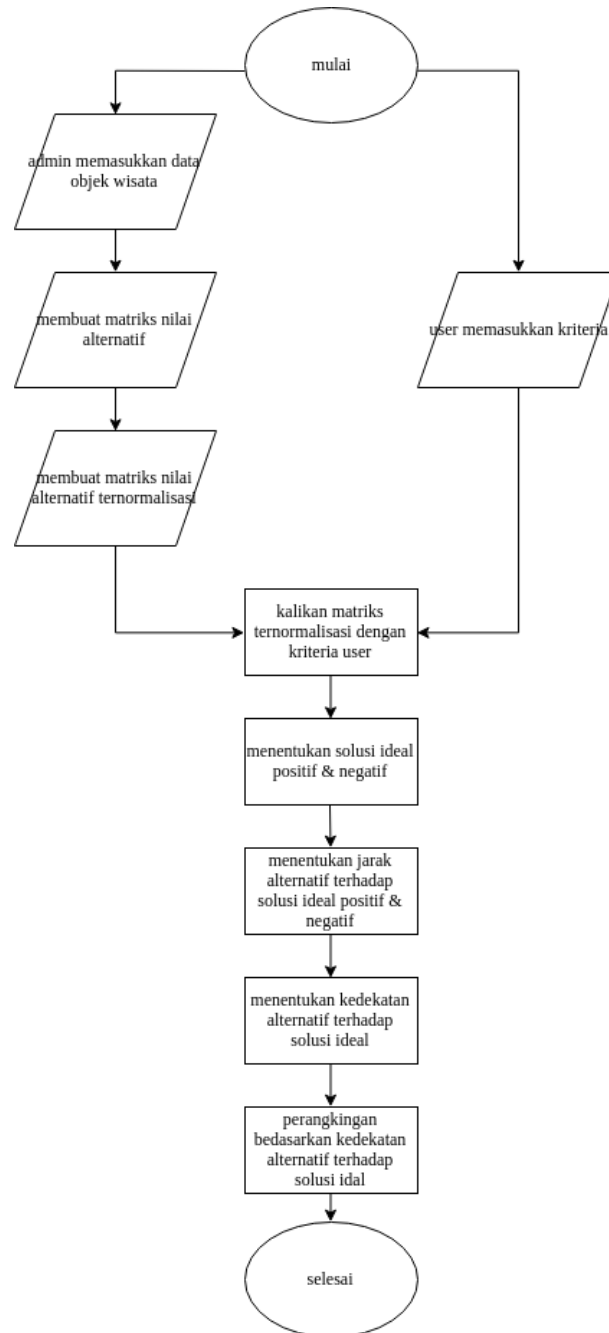
dengan transportasi. Transportasi yang di maksud dalam kriteria ini yaitu jalan kaki, sepeda motor, mobil, bus dan transportasi umum. Semakin mudah sebuah objek wisata dijangkau maka semakin besar nilai kriteria nya. Nilai kriteria transportasi disajikan dalam Tabel 4.7.

Tabel 4.7 Tabel Kriteria Fasilitas

Kriteria	Tipe	Keterangan Nilai	Nilai
Fasilitas	Benefit	Ada transportasi umum yang menuju objek wisata	5
		Dapat ditempuh dengan bus	4
		Dapat ditempuh dengan mobil	3
		Dapat ditempuh dengan sepeda motor	2
		Harus ditempuh dengan jalan kaki	1

4.1.7. Analisis Perhitungan Model TOPSIS

Metode *TOPSIS* memiliki konsep dimana alternatif yang terpilih merupakan alternative terbaik yang memiliki jarak terpendek dari solusi ideal positif dan jarak terjauh dari solusi ideal negatif. Dalam sistem ini, alur perhitungan metode *TOPSIS* disajikan dalam Gambar 4.1.



Gambar 4.1 Flowhart Penghitungan Metode TOPSIS dalam Sistem

1. Admin memasukkan data objek wisata

Langkah pertama adalah admin memasukkan data objek wisata yang ada di Daerah Istimewa Yogyakarta secara detail.

2. Membuat matriks nilai alternatif

Langkah selanjutnya adalah sistem menghitung dan membuat matriks nilai alternatif.

3. Membuat matriks nilai ternormalisasi

Setelah matriks nilai alternatif berhasil dibuat, selanjutnya adalah membuat matriks nilai ternormalisasi. Nilai ternormalisasi (r) pada alternatif ke- i , kriteria ke- j adalah hasil dari nilai kriteria alternatif(x) dibagi dengan akar dari jumlah kuadrat seluruh nilai kriteria ke- j pada setiap alternatif. Perhitungan matriks nilai ternormalisasi dijelaskan pada Rumus 4.1.

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}} \quad (4.1)$$

4. *User* menentukan kriteria

User memasukkan seberapa pentingkah 7 kriteria yang disediakan sistem dalam skala 1 sampai 5.

5. Membuat matriks ternormalisasi terbobot

Setelah membuat matriks ternormalisasi, langkah selanjutnya membuat matriks ternormalisasi terbobot dengan mengalikan matriks ternormalisasi dengan bobot kriteria yang dimasukkan *user*. Nilai ternormalisasi terbobot (y) pada alternatif ke- i , kriteria ke- j adalah hasil dari nilai ternormalisasi(r) alternatif dikali dengan kriteria yang dimasukkan *user*(w). Perhitungan matriks nilai ternormalisasi terbobot

dijelaskan pada Rumus 4.2.

$$y_{ij} = w_i r_{ij} \quad (4.2)$$

6. Menentukan nilai solusi ideal negatif dan positif

Langkah selanjutnya adalah menentukan nilai solusi ideal positif dan negatif. Nilai solusi ideal positif dijelaskan pada Rumus 4.3.

$$A^+ = (y1^+, y2^+, y3^+, \dots, yn^+) \quad (4.3)$$

Nilai solusi ideal positif adalah nilai *max* kriteria dari seluruh alternatif jika kriteria bersifat *benefit* dan *min* kriteria jika kriteria bersifat *cost*.

Nilai solusi ideal negatif dijelaskan pada Rumus 4.4.

$$A^- = (y1^-, y2^-, y3^-, \dots, yn^-) \quad (4.4)$$

Nilai solusi ideal negatif adalah nilai *max* kriteria dari seluruh alternatif jika kriteria bersifat *cost* dan *min* kriteria jika kriteria bersifat *benefit*.

7. Menghitung jarak alternatif terhadap solusi ideal

Setelah menentukan nilai solusi ideal positif dan negatif, langkah selanjutnya adalah menghitung jarak alternatif terhadap nilai solusi ideal positif dan negatif.

Jarak solusi ideal positif suatu alternatif adalah akar dari jumlah kuadrat setiap kriteria dikurangi dengan nilai solusi ideal positif kriteria tersebut. Pehitungan jarak solusi ideal positif dijelaskan pada Rumus 4.5.

$$D_i^+ = \sqrt{\sum_{i=1}^n (y_i^+ - y_{ij})}, i = 1, 2, 3, \dots, m \quad (4.5)$$

Jarak solusi ideal negatif suatu alternatif adalah akar dari jumlah kuadrat setiap kriteria dikurangi dengan nilai solusi ideal negatif kriteria tersebut. Pehitungan jarak

solusi ideal positif dijelaskan pada Rumus 4.6.

$$D_i^- = \sqrt{\sum_{j=1}^n (y_{ij} - y_i^-)}, i = 1, 2, 3, \dots, m \quad (4.6)$$

8. Menghitung kedekatan alternatif terhadap solusi ideal

Kedekatan alternatif terhadap solusi ideal adalah nilai jarak solusi ideal negatif dibagi dengan jumlah nilai jarak ideal positif dan nilai jarak solusi ideal negatif alternatif tersebut. Kedekatan alternatif terhadap solusi ideal dijelaskan pada Rumus 4.7.

$$V = \frac{D_i^-}{D_i^+ + D_i^-}, i = 1, 2, 3, \dots, m \quad (4.7)$$

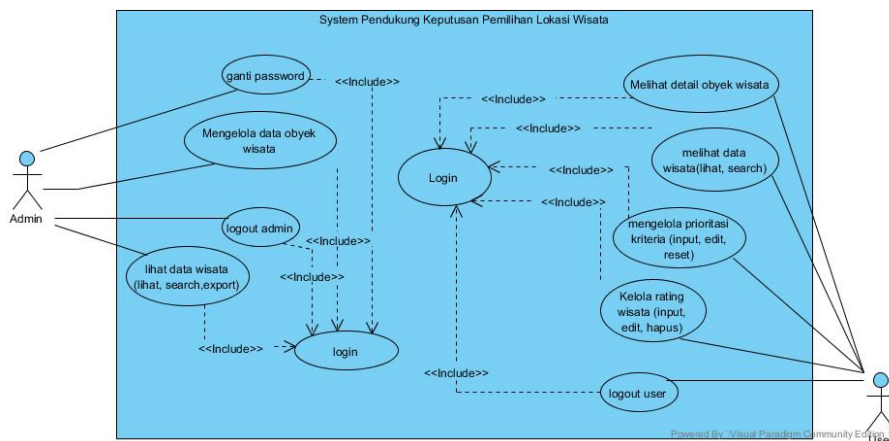
9. Me-rangking alternatif berdasarkan kedekatannya terhadap solusi ideal

Langkah terakhir adalah mengurutkan alternatif berdasarkan kedekatannya terhadap solusi ideal yang telah dihitung.

4.2. Perancangan Sistem

4.2.1. Use Case Diagram

Use case diagram adalah suatu diagram yang menggambarkan interaksi antara sistem dengan lingkungan yang berada di luar sistem. Atau dengan kata lain *use case* memvisualisasikan siapa pengguna sistem dan bagaimana pengguna berinteraksi dengan sistem tersebut. Dalam penelitian ini diagram *use case* dijelaskan pada Gambar 4.2.



Gambar 4.2 Use Case Diagram Sistem

Pada Gambar 4.2 dijelaskan admin dapat menjalankan beberapa fungsi diantaranya mengganti *password*, melihat data objek wisata, *log out*, dan mengelola (menambah, menghapus, mengubah) data objek wisata. Namun sebelumnya admin harus *login* dulu untuk dapat masuk ke dalam sistem.

Sementara, *user* dapat melihat seluruh objek wisata, melakukan pencarian objek wisata, menentukan kriteria yang mereka inginkan dan melihat detail objek wisata. Seperti admin, *user* harus *login* terlebih dahulu. Setelah memasukkan kriteria yang diinginkan, maka *user* akan mendapat rekomendasi objek wisata.

4.2.2. Activity Diagram

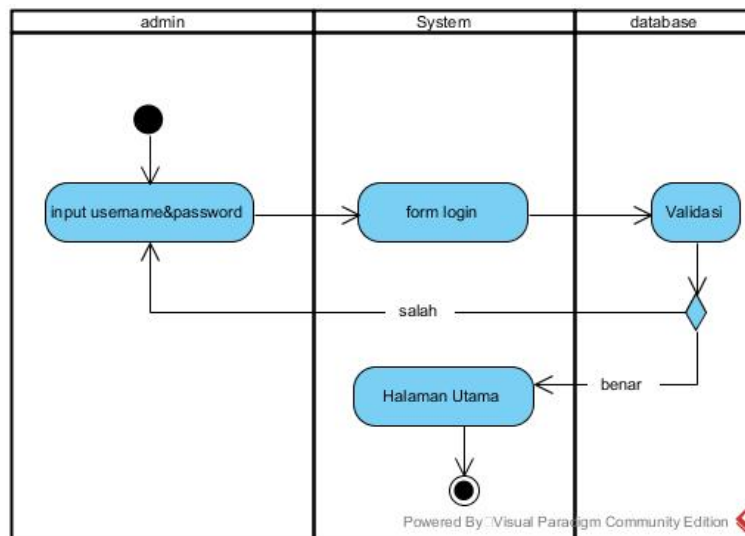
Setelah menemukan perilaku (*behavior*) apa saja yang dilakukan oleh setiap aktor (pengguna), yaitu admin dan *user* di dalam sistem dengan menggunakan *use case diagram*, tahap selanjutnya yaitu mengubah setiap aktivitas aktor kedalam suatu *activity diagram*, dan pada tahap ini akan didapatkan hasil alur yang terjadi ketika aktivitas itu berjalan.

1. Activity Diagram Admin Login

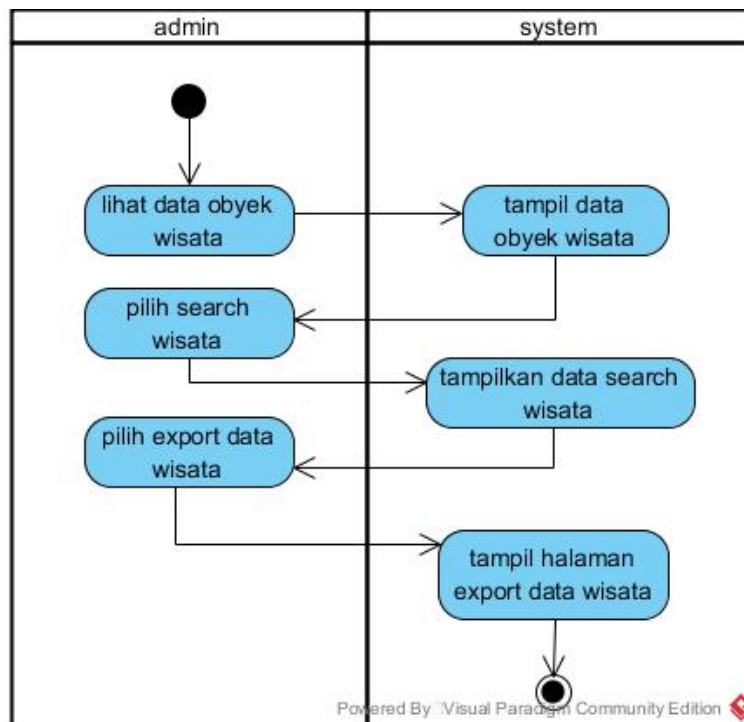
Diagram aktivitas ini memodelkan alur sistem saat admin akan *login*. Saat menampilkan halaman *login*, admin diharuskan memasukkan *username* dan *password* agar sistem dapat memverifikasinya. Jika berhasil, aktor akan diarahkan ke halaman *dashboard* admin. Sebaliknya jika gagal, aktor akan diarahkan kembali ke halaman *login*. Diagram aktivitas fungsi ini dijelaskan pada Gambar 4.3.

2. Activity Diagram Admin Lihat Seluruh Objek Wisata

Diagram aktivitas ini memodelkan alur sistem admin pada fungsi melihat seluruh objek wisata. Aktor dapat melihat seluruh objek wisata yang sudah disimpan di *database*. Terdapat beberapa subfungsi dalam fungsi ini diantaranya cari objek wisata dan *export* data objek wisata. Diagram aktivitas fungsi ini dijelaskan pada Gambar 4.4.



Gambar 4.3 Activity Diagram Admin Login



Gambar 4.4 Activity Diagram Admin Lihat Seluruh Objek Wisata

3. Activity Diagram Admin Mengelola Data Objek Wisata

Diagram aktivitas ini memodelkan alur sistem admin pada fungsi mengelola objek wisata. Admin dapat melakukan beberapa fungsi untuk mengelola data objek wisata diantaranya menambah data, memperbarui data, dan menghapus data. Saat menambah atau memperbarui data, admin diharuskan untuk mengisi seluruh detail objek wisata. Jika tidak diisi, maka fungsi tidak akan bisa dijalankan dan kembali kehalaman selanjutnya dengan pesan gagal. Saat menghapus data, akan muncul *popup* untuk mengkonfirmasi apakah data akan benar-benar dihapus. Diagram aktivitas fungsi ini dijelaskan pada Gambar 4.5.

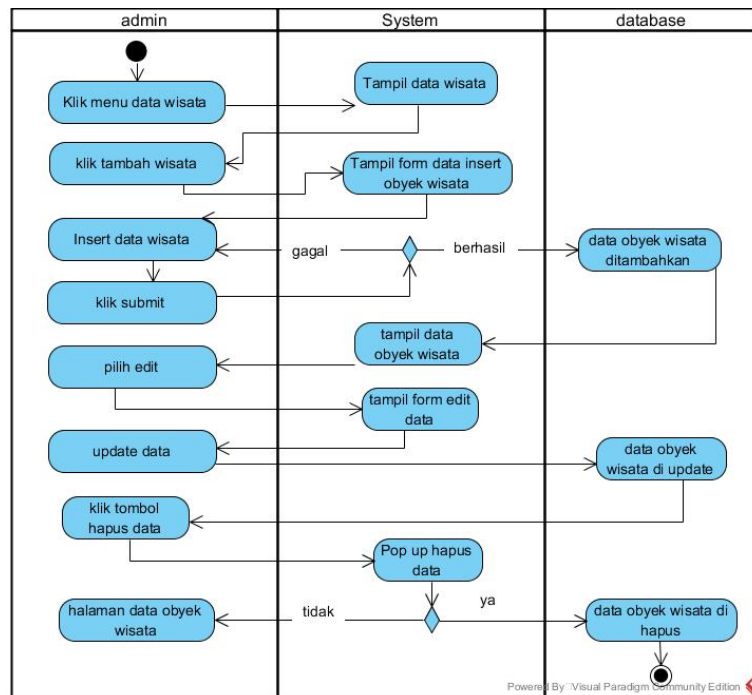
4. Activity Diagram Admin Mengganti Password

Diagram aktivitas ini memodelkan alur sistem admin pada fungsi mengganti *password*. Admin diharuskan untuk memasukkan *password* lama dan *password* baru.

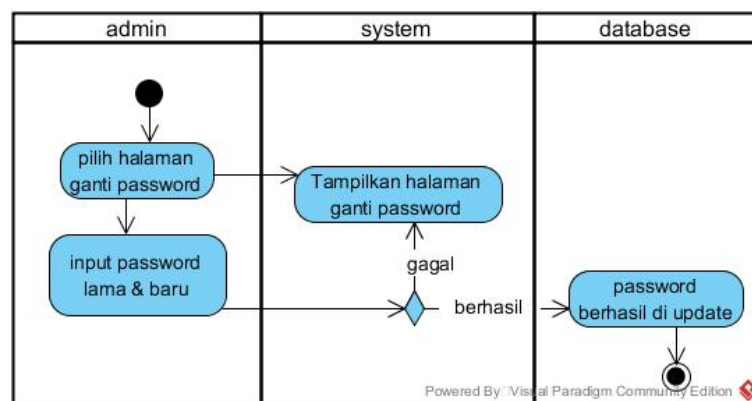
Jika terjadi kesalahan saat mengganti *password* maka akan kembali ke halaman ini dengan pesan gagal. Diagram aktivitas fungsi ini dijelaskan pada Gambar 4.6.

5. Activity Diagram Admin Log Out

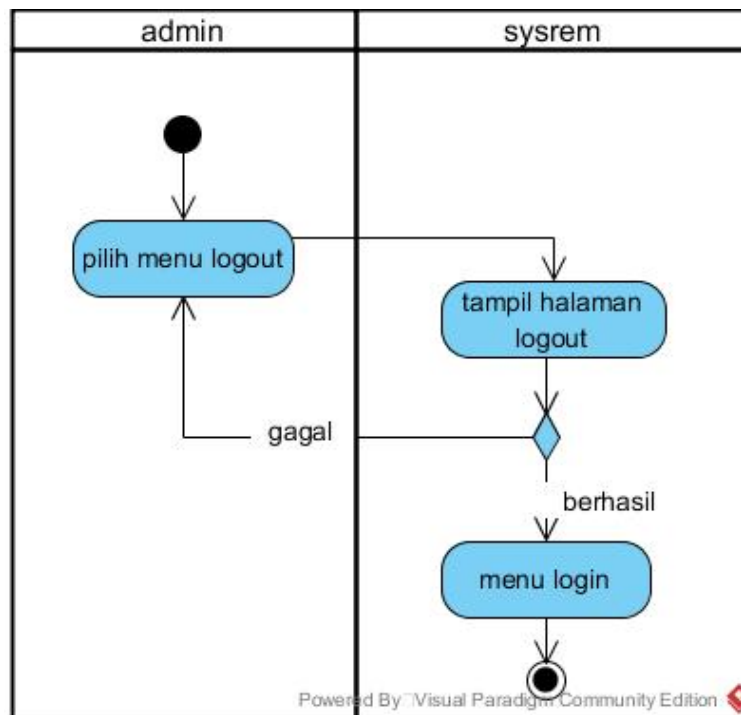
Diagram aktiivitas ini memodelkan alur sistem admin pada fungsi *log out*. Diagram aktivitas fungsi ini dijelaskan pada Gambar 4.7.



Gambar 4.5 Activity Diagram Admin Mengelola Data Objek Wisata



Gambar 4.6 Activity Diagram Admin Mengganti Password



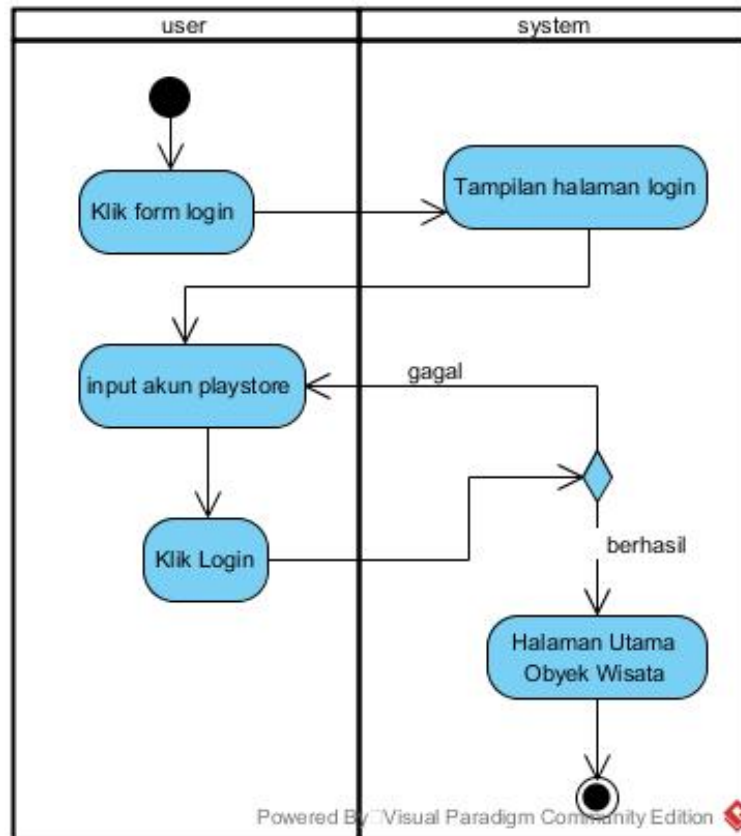
Gambar 4.7 Activity Diagram Admin Log Out

6. Activity Diagram User Login

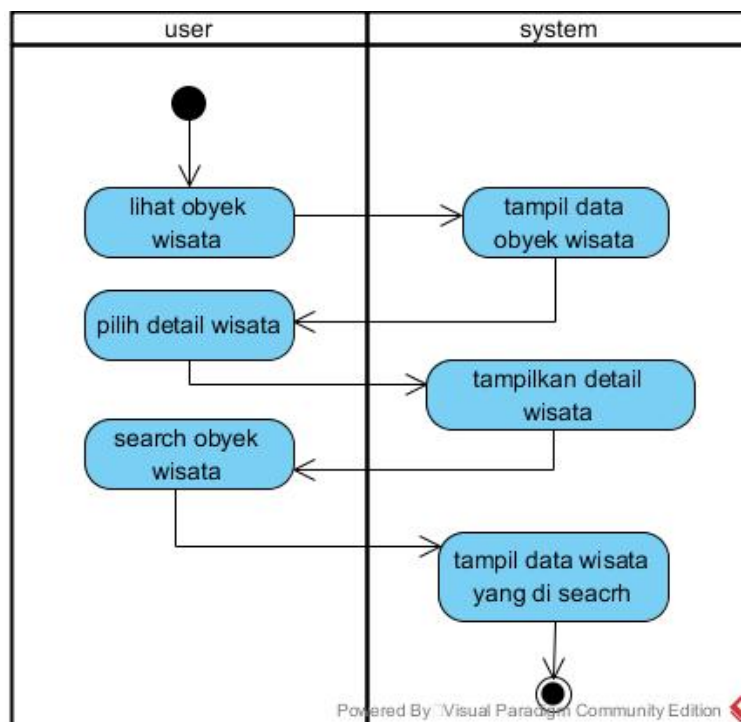
Diagram ini memodelkan saat *user* pertama kali membuka *mobile apps* sistem. *User* diharuskan memilih salah satu *google account* yang terdaftar pada *smartphone*-nya sebelum menggunakan aplikasi. Jika berhasil maka *user* diarahkan ke halaman utama. Diagram aktivitas fungsi ini dijelaskan pada Gambar 4.8.

7. Activity Diagram User Melihat Semua Objek Wisata

Diagram ini memodelkan saat *user* berhasil *login*, *user* akan diarahkan ke halaman utama. Halaman ini akan menampilkan seluruh data objek wisata pada *user*. Dalam fungsi ini ada satu subfungsi, yaitu *user* dapat mencari objek wisata. Diagram aktivitas fungsi ini dijelaskan pada Gambar 4.9.



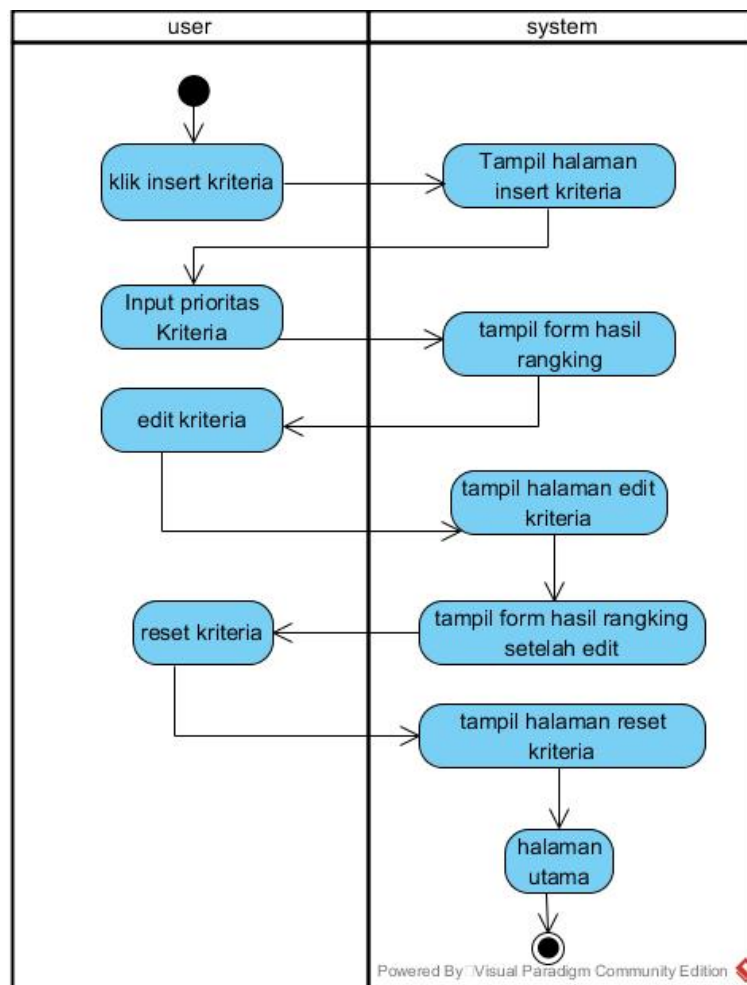
Gambar 4.8 Activity Diagram User Login



Gambar 4.9 Activity Diagram User Melihat Semua Obyek Wisata

8. Activity Diagram User Memasukkan Kriteria

Diagram ini memodelkan saat *user* ingin mencari rekomendasi objek wisata dengan aplikasi ini. Akan muncul halaman dimana *user* harus memasukkan seberapa pentingkah mereka mempertimbangkan 7 kriteria yang disediakan dalam skala 1 sampai 5. *User* juga dapat mengatur ulang (*reset*) kriteria yang telah mereka masukkan. Diagram aktivitas fungsi ini dijelaskan pada Gambar 4.10.

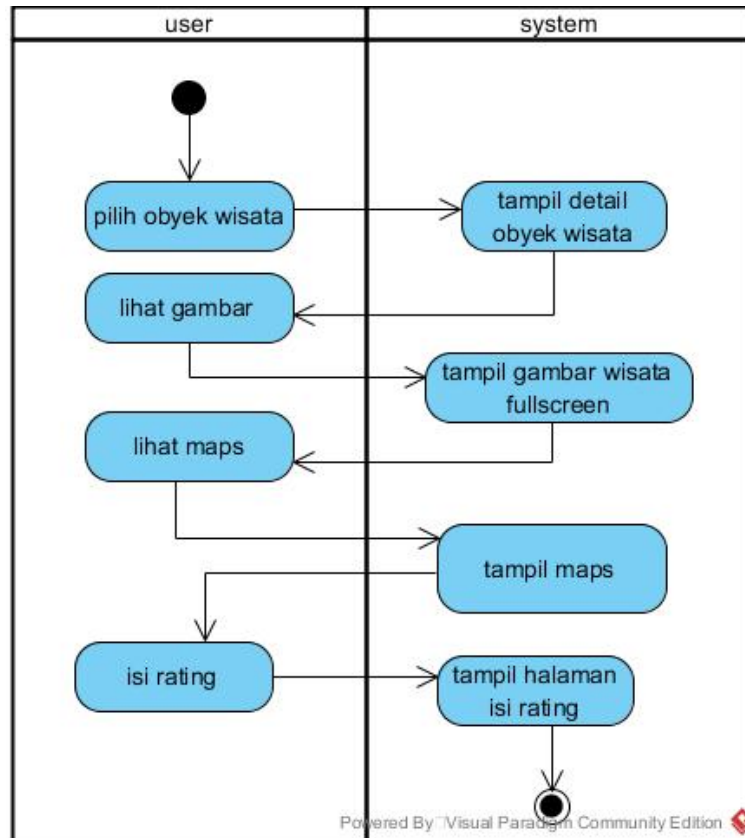


Gambar 4.10 Activity Diagram User Memasukkan Kriteria

9. Activity Diagram User Melihat Detail Objek Wisata

Diagram ini memodelkan saat *user* memilih salah satu objek wisata dari halaman utama. *User* dapat melihat detail objek wisata yang dipilih. Selain itu, *user* juga

dapat membuka *google maps* untuk mengetahui rute terbaik ke arah objek wisata yang disediakan oleh *google* dan dapat melihat gambar-gambar galeri objek wisata secara *fullscreen*. Diagram aktivitas fungsi ini dijelaskan pada Gambar 4.11.



Gambar 4.11 Activity Diagram User Melihat Detail Objek Wisata

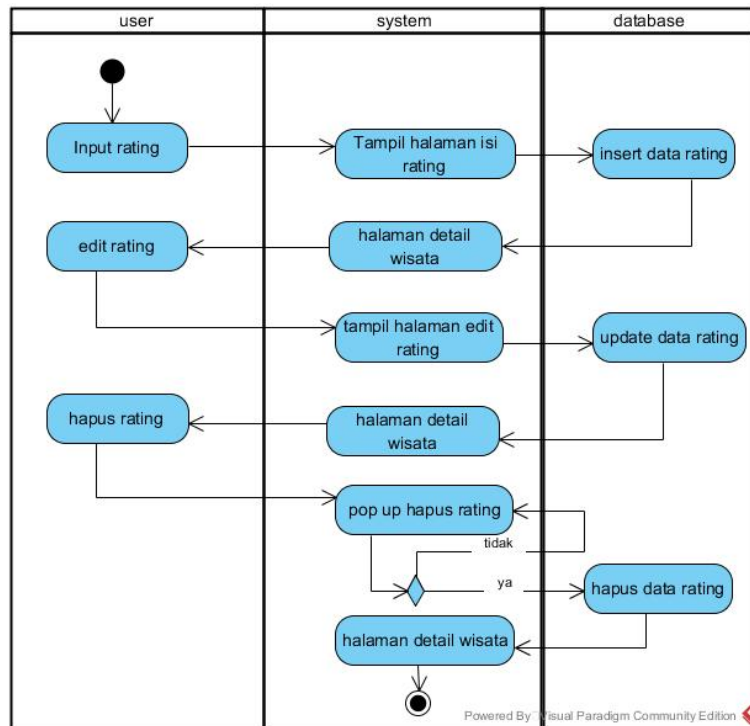
10. Activity Diagram User Mengelola Ulasan(Rating)

Diagram ini memodelkan *user* saat mengelola ulasan terhadap suatu objek wisata. Saat *user* akan menambah atau memperbarui ulasan, akan muncul *pop up* halaman untuk menulis ulasan yang harus diisi *user*. Saat akan menghapus ulasan, akan muncul pula *pop up* untuk konfirmasi apakah ingin benar-benar menghapus ulasan mereka. Diagram aktivitas fungsi ini dijelaskan pada Gambar 4.12.

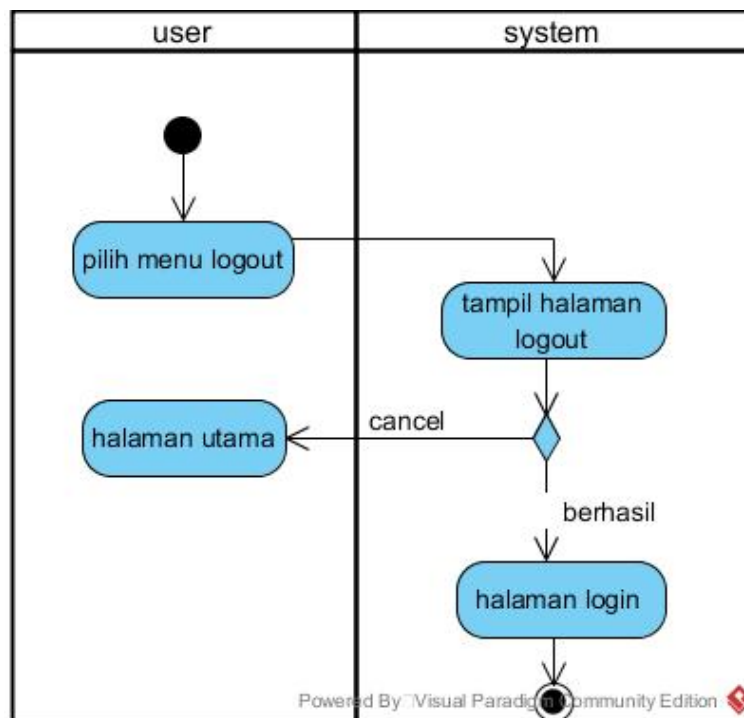
11. Activity Diagram User Log Out

Diagram ini memodelkan saat *user* ingin *log out* atau mengganti akun yang mereka

gunakan pada aplikasi. Diagram aktivitas fungsi ini dijelaskan pada Gambar 4.13.



Gambar 4.12 Activity Diagram User Mengelola Ulasan(Rating)

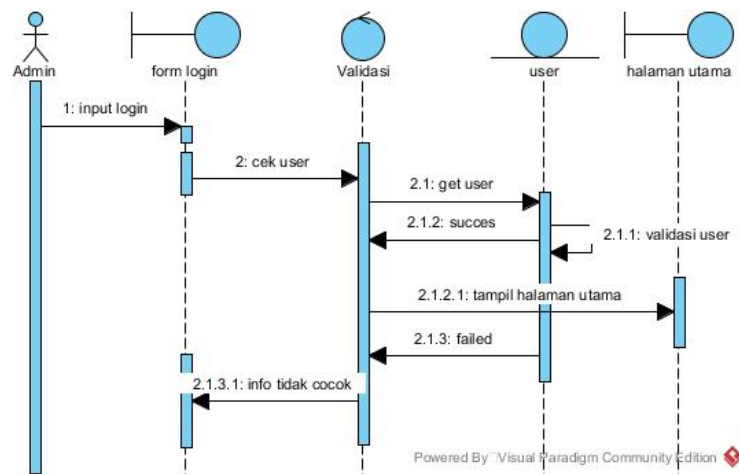


Gambar 4.13 Activity Diagram User Log Out

4.2.3. Sequence Diagram

1. Sequence Diagram Admin Login

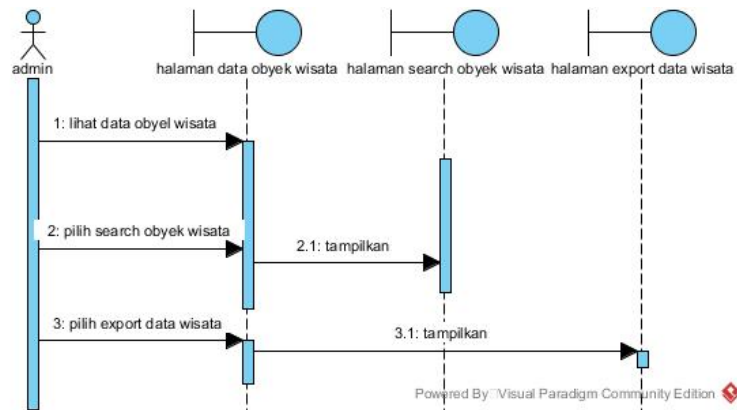
Diagram ini menjelaskan apa saja yang terjadi jika admin ingin masuk ke sistem. Pertama, akan muncul halaman *login* yang berisi form yang harus diisi oleh admin. Setelah mengisi form, maka sistem akan memvalidasi data yang dimasukkan admin ke database. Jika data valid, maka admin akan diarahkan ke halaman utama. Dan jika gagal akan menampilkan kembali halaman untuk *login* dengan pesan gagal. Diagram *sequence* fungsi ini dijelaskan pada Gambar 4.14.



Gambar 4.14 Sequence Diagram Admin Login

2. Sequence Diagram Admin Lihat Seluruh Objek Wisata

Diagram ini menjelaskan apa saja yang terjadi saat admin melihat data objek wisata. Sistem akan memuat data dari *database* dan menampilkannya. Admin juga dapat mencari objek wisata dan dapat meng-*export* data objek wisata untuk laporan. Diagram *sequence* fungsi ini dijelaskan pada Gambar 4.15.



Gambar 4.15 *Sequence Diagram Admin Lihat Seluruh Objek Wisata*

3. *Sequence Diagram Admin Mengelola Data Objek Wisata*

Diagram ini menjelaskan apa saja yang terjadi saat admin mengelola data objek wisata. Ada 3 fungsi utama admin saat mengelola data objek wisata yaitu menambah data, memperbarui data, dan menghapus data. Saat menambah dan memperbarui data objek wisata, admin harus mengisi semua detail objek wisata. Jika ada data yang kosong, maka sistem akan menampilkan pesan bahwa seluruh data harus diisi. Sementara untuk menghapus data, sistem akan langsung menghapus data dari *database*. Diagram *sequence* fungsi ini dijelaskan pada Gambar 4.16.

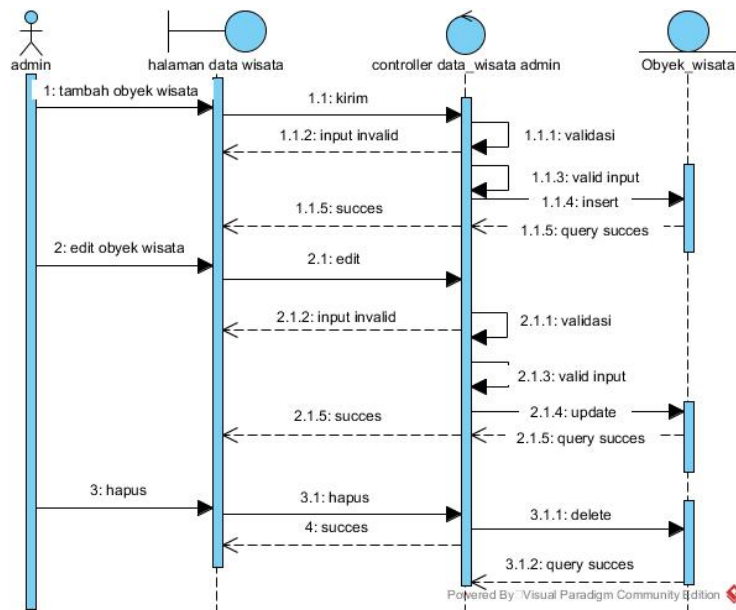
4. *Sequence Diagram Admin Mengganti Password*

Diagram ini menjelaskan apa saja yang terjadi saat admin akan mengganti *password*. Akan muncul halaman dimana admin harus mengisi *password* lama, *password* baru dan konfirmasi *password* baru. Jika semua kondisi terpenuhi maka sistem akan menampilkan pesan berhasil dan menyimpan *password* baru. Diagram *sequence* fungsi ini dijelaskan pada Gambar 4.17.

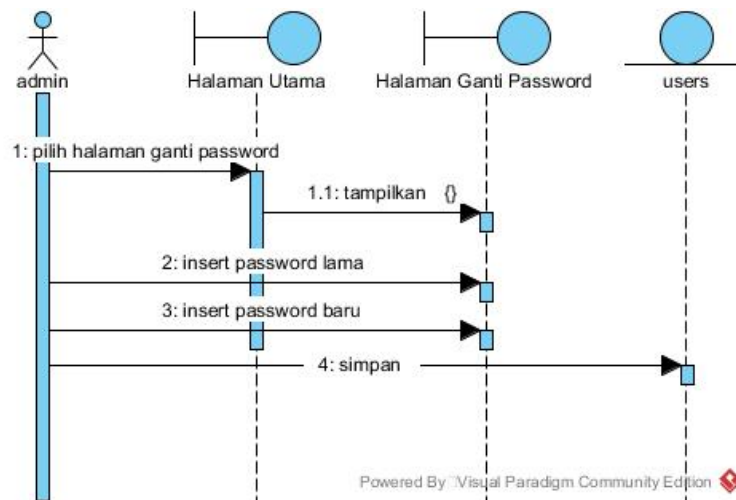
5. *Sequence Diagram Admin Log Out*

Diagram ini menjelaskan saat admin *log out*. Saat *log out* berhasil maka sistem

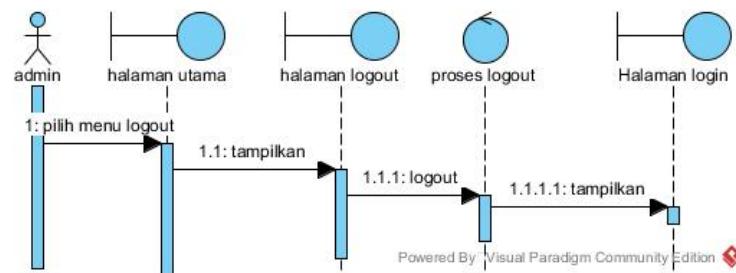
akan mengarahkan admin ke halaman *login* awal. Diagram *sequence* fungsi ini dijelaskan pada Gambar 4.18.



Gambar 4.16 Sequence Diagram Admin Mengelola Data Objek Wisata



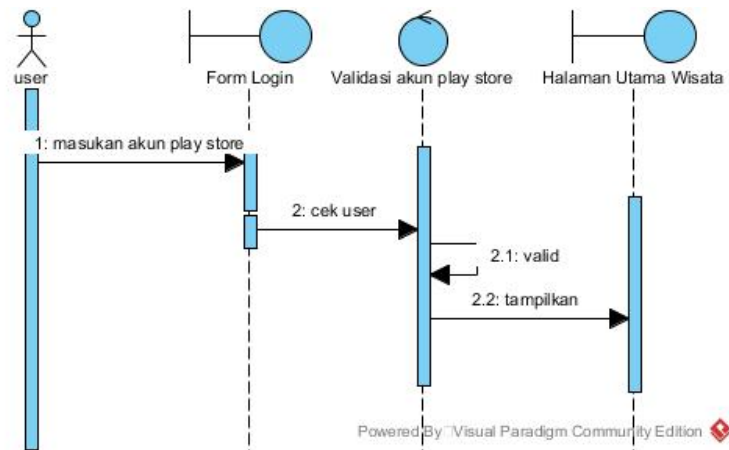
Gambar 4.17 Sequence Diagram Admin Mengganti Password



Gambar 4.18 Sequence Diagram Admin Log Out

6. Sequence Diagram User Login

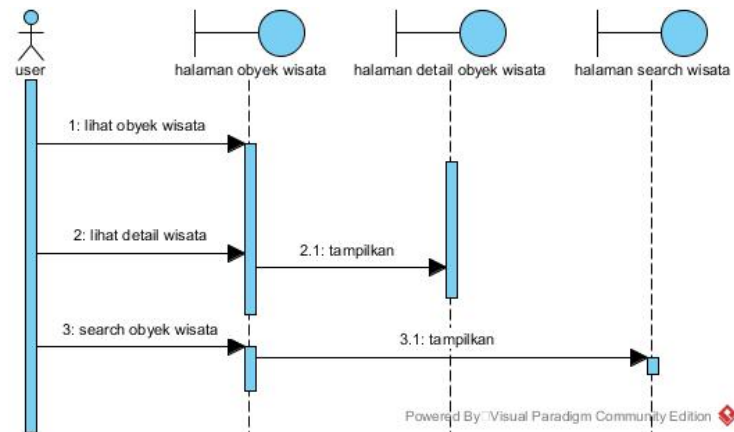
Diagram ini menjelaskan apa saja yang terjadi saat *user login*, *user* harus memilih salah satu *google account* yang terdaftar pada *smartphone* mereka. Saat berhasil, *user* akan diarahkan ke halaman utama. Diagram *sequence* fungsi ini dijelaskan pada Gambar 4.19.



Gambar 4.19 Sequence Diagram User Login

7. Sequence Diagram User Melihat Semua Objek Wisata

Diagram ini menjelaskan saat *user* melihat semua objek wisata. Sistem akan menampilkan semua objek wisata kepada *user*, saat *user* menekan salah satu objek wisata maka *user* akan diarahkan ke halaman detail objek wisata. *User* juga dapat melakukan pencarian objek wisata. Diagram *sequence* fungsi ini dijelaskan pada Gambar 4.20.



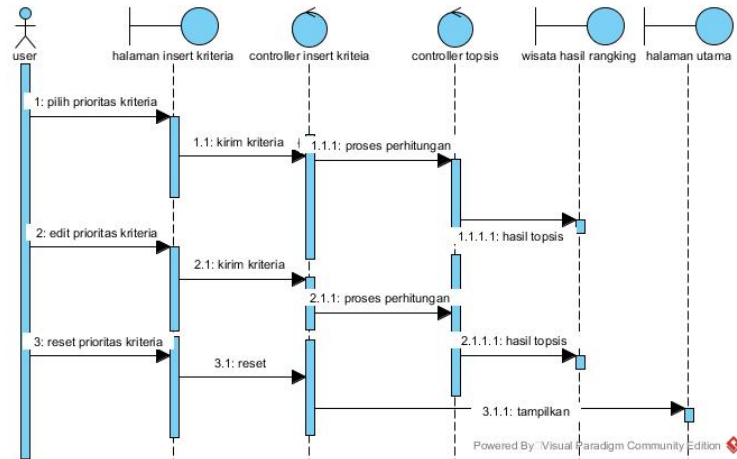
Gambar 4.20 *Sequence Diagram User Melihat Semua Objek Wisata*

8. *Sequence Diagram User Memasukkan Kriteria*

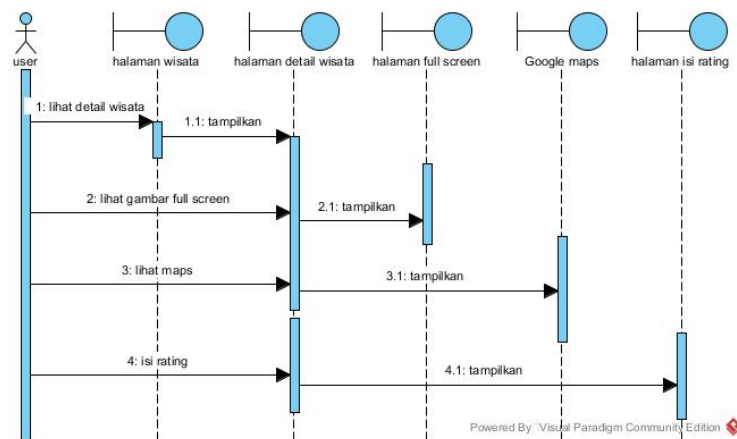
Diagram ini menjelaskan saat *user* ingin mencari rekomendasi objek wisata dengan sistem. *User* harus mengisi kriteria yang disediakan oleh sistem. Setelah mengisi kriteria yang diinginkan, sistem akan melakukan perhitungan untuk mencari rekomendasi objek wisata. Saat selesai melakukan perhitungan, sistem akan menampilkan urutan atau hasil perhitungan *TOPSIS* kepada *user*. Saat *user* me-reset kriteria, maka sistem akan menampilkan seluruh objek wisata kepada *user*. Diagram *sequence* fungsi ini dijelaskan pada Gambar 4.21.

9. *Sequence Diagram User Melihat Detail Objek Wisata*

Diagram ini menjelaskan saat *user* melihat detail objek wisata. Dalam halaman detail objek wisata, terdapat beberapa fungsi yaitu melihat gambar galeri, membuka *google maps*, dan mengelola ulasan *user* pada suatu objek wisata. Diagram *sequence* fungsi ini dijelaskan pada Gambar 4.22.



Gambar 4.21 Sequence Diagram User Memasukkan Kriteria



Gambar 4.22 Sequence Diagram User Melihat Detail Objek Wisata

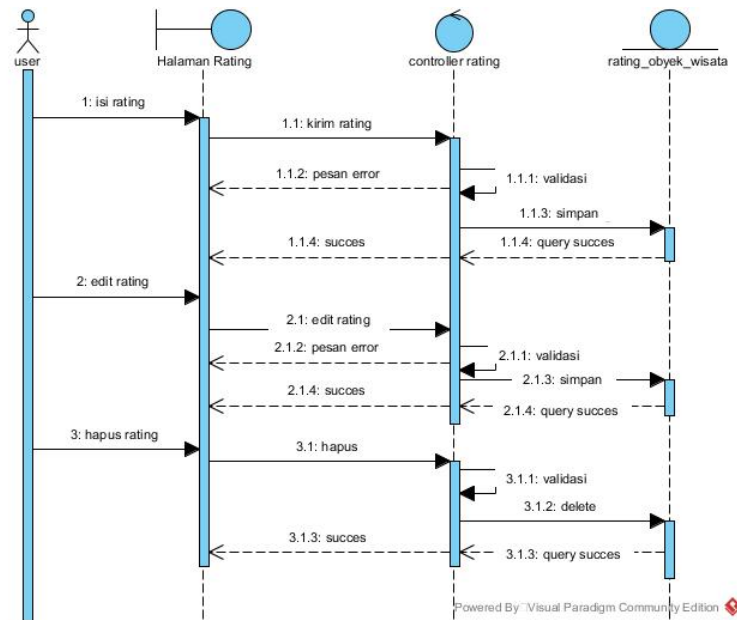
10. Sequence Diagram User Mengelola Ulasan(Rating)

Diagram ini menjelaskan saat *user* mengelola ulasan mereka pada objek wisata. Ada tiga aksi yang dapat *user* lakukan yaitu menambah ulasan, memperbarui ulasan dan menghapus ulasan. Saat menambah atau memperbarui ulasan, sistem akan mengirim data ke *database*. Jika berhasil maka akan menampilkan berhasil dan sebaliknya Saat akan menghapus ulasan, akan muncul *pop up* terlebih dahulu. Diagram *sequence* fungsi ini dijelaskan pada Gambar 4.23.

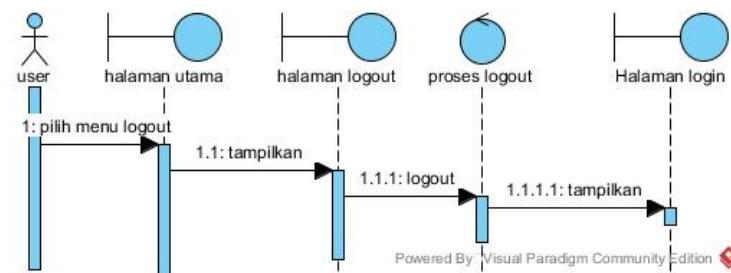
11. Sequence Diagram User Log Out

Saat *log out*, jika berhasil maka *user* akan diarahkan kembali ke halaman *login*.

Diagram *sequence* fungsi ini dijelaskan pada Gambar 4.24.



Gambar 4.23 Sequence Diagram User Mengelola Ulasan(Rating)

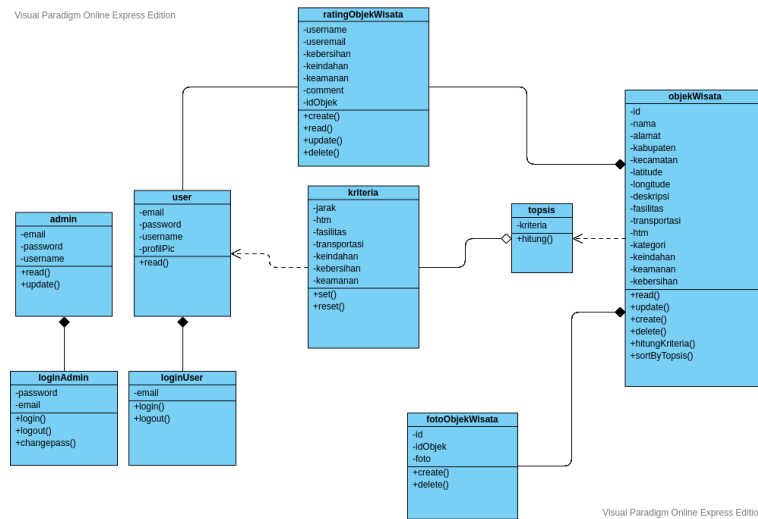


Gambar 4.24 Sequence Diagram User Log Out

4.2.4. Class Diagram

Setelah diagram *use case*, *activity*, dan *sequence* selesai dibuat, untuk proses selanjutnya dilanjutkan dengan pembuatan *class diagram*. Setelah merancang *behavior* dan *interactions diagram*, dilanjutkan dengan merancang *structure diagram* yaitu *class diagram*. Diagram yang digunakan untuk menggambarkan struktur statis yang terdapat dalam sistem, kelas-kelas yang akan dibangun merepresentasikan sesuatu yang nantinya

akan ditangani oleh sistem, dan menunjukkan hubungan antara mereka. Diagram ini difungsikan untuk mengetahui arsitektur dari sistem yang akan dibangun. Dalam penelitian ini perancangan *class diagram* diperlihatkan seperti pada Gambar 4.25.



Gambar 4.25 Class Diagram

4.2.5. Struktur Tabel

1. Tabel Objek Wisata

Tabel objek wisata ini adalah tabel untuk menyimpan data objek wisata di Daerah Istimewa Yogyakarta. Tabel ini memiliki 17 *fields* antara lain id, nama, kabupaten, kecamatan, alamat, deskripsi, latitude, longitude, kategori, htm, fasilitas, transportasi, kebersihan, keindahan, keamanan, *created_at*, *updated_at*. Struktur tabel ini disajikan pada Tabel 4.8.

Tabel 4.8 Struktur Tabel Objek Wisata

No	Nama <i>Field</i>	Tipe	Keterangan
1	id	<i>int</i> (10)	<i>Primary key</i>
2	nama	<i>varchar</i> (50)	Nama objek wisata

Tabel 4.9 Struktur Tabel Objek Wisata (Lanjutan)

No	Nama <i>Field</i>	Tipe	Keterangan
3	kabupaten	<i>varchar(25)</i>	Kabupaten lokasi objek wisata. Nilai pada <i>field</i> ini diambil dari tabel kabupaten
4	kecamatan	<i>varchar(25)</i>	Kecamatan lokasi objek wisata. Nilai pada <i>field</i> ini diambil dari tabel kecamatan
5	alamat	<i>varchar(200)</i>	Alamat lengkap objek wisata
6	deskripsi	<i>text</i>	Deskripsi yang menggambarkan objek wisata untuk menarik minat wisatawan
7	<i>latitude</i>	<i>varchar(20)</i>	Koordinat <i>latitude</i> objek wisata
8	<i>longitude</i>	<i>varchar(20)</i>	Koordinat <i>longitude</i> objek wisata
9	kategori	<i>enum</i> (Alam, Pantai, Budaya, Museum, Minat Khusus)	Kategori objek wisata yang terbagi menjadi 5 kategori sesuai dengan halaman web Dinas Pariwisata DIY
10	htm	<i>int(11)</i>	Harga tiket masuk objek wisata

Tabel 4.10 Struktur Tabel Objek Wisata (Lanjutan)

No	Nama <i>Field</i>	Tipe	Keterangan
11	fasilitas	<i>varchar</i> (50)	Fasilitas apa saja yang ada pada objek wisata
12	transportasi	<i>enum</i> (1, 2, 3, 4, 5)	<i>Field</i> untuk menyimpan seberapa mudah akses transportasi ke objek wisata
13	kebersihan	<i>double</i> (2,1)	Nilai kriteria kebersihan objek wisata yang diambil dari rata-rata nilai kebersihan pada tabel rating objek wisata
14	keamanan	<i>double</i> (2,1)	Nilai kriteria keamanan objek wisata yang diambil dari rata-rata nilai keamanan pada tabel rating objek wisata
15	keindahan	<i>double</i> (2,1)	Nilai kriteria keindahan objek wisata yang diambil dari rata-rata nilai keindahan pada tabel rating objek wisata

2. Tabel Rating Objek Wisata

Tabel rating objek wisata ini adalah tabel untuk menyimpan data penilaian oleh

user terhadap suatu objek wisata di Daerah Istimewa Yogyakarta. Tabel ini memiliki 10 *fields* antara lain *id*, *id_objek*, *username*, *useremail*, *kebersihan*, *keamanan*, *keindahan*, *comment*, *created_at*, *updated_at*. Struktur tabel rating objek wisata disajikan dalam Tabel 4.11.

Tabel 4.11 Struktur Tabel Rating Objek Wisata

No	Nama <i>Field</i>	Tipe	Keterangan
1	<i>id</i>	<i>int(10)</i>	<i>Primary key</i>
2	<i>id_objek</i>	<i>int(11)</i>	<i>Foreign key</i> yang me-relasi kan tabel ini dengan tabel objek wisata
3	<i>username</i>	<i>varchar(30)</i>	<i>Username user</i> yang memberikan rating
4	<i>useremail</i>	<i>varchar(30)</i>	<i>Useremail user</i> yang memberikan rating
5	<i>kebersihan</i>	<i>int(11)</i>	Nilai kebersihan yang diberikan oleh <i>user</i> terhadap suatu objek wisata
6	<i>keamanan</i>	<i>int(11)</i>	Nilai keamanan yang diberikan oleh <i>user</i> terhadap suatu objek wisata
7	<i>keindahan</i>	<i>int(11)</i>	Nilai keindahan yang diberikan oleh <i>user</i> terhadap suatu objek wisata

Tabel 4.12 Struktur Tabel Rating Objek Wisata (Lanjutan)

No	Nama <i>Field</i>	Tipe	Keterangan
8	<i>comment</i>	<i>varchar(255)</i>	Komentar yang diberikan <i>user</i> terhadap suatu objek wisata

3. Tabel Foto Objek Wisata

Tabel foto objek wisata ini adalah tabel untuk menyimpan data foto objek wisata di Daerah Istimewa Yogyakarta. Tabel ini memiliki 5 *fields* antara lain *id*, *id_objek*, *foto*, *created_at*, *updated_at*. Struktur tabel foto objek wisata disajikan dalam Tabel 4.13.

Tabel 4.13 Struktur Tabel Foto Objek Wisata

No	Nama <i>Field</i>	Tipe	Keterangan
1	<i>id</i>	<i>int(10)</i>	<i>Primary key</i>
2	<i>id_objek</i>	<i>int(11)</i>	<i>Foreign key</i> yang me-relasi kan tabel ini dengan tabel objek wisata
3	<i>foto</i>	<i>varchar(100)</i>	Nama file foto

4. Tabel Kabupaten

Tabel kabupaten ini adalah tabel untuk menyimpan data kabupaten di Daerah Istimewa Yogyakarta. Data kabupaten ini digunakan untuk memudahkan admin memasukkan kabupaten lokasi objek wisata. Tabel ini memiliki 4 *fields* antara lain *id*, *nama_kabupaten*, *created_at*, *updated_at*. Struktur tabel kabupaten disajikan dalam Tabel 4.14.

Tabel 4.14 Struktur Tabel Kabupaten

No	Nama <i>Field</i>	Tipe	Keterangan
1	<i>id</i>	<i>int(10)</i>	<i>Primary key</i>
2	<i>nama_kabupaten</i>	<i>varchar(255)</i>	Nama kabupaten

5. Tabel Kecamatan

Tabel kecamatan ini adalah tabel untuk menyimpan data kecamatan di Daerah Istimewa Yogyakarta. Data kecamatan ini digunakan untuk memudahkan admin memasukkan kecamatan lokasi objek wisata. Tabel ini memiliki 4 *fields* antara lain *id*, *id_kab*, *nama_kecamatan*, *created_at*, *updated_at*. Struktur tabel kecamatan disajikan dalam tabel 4.15.

6. Tabel *Users*

Tabel *users* ini adalah tabel untuk menyimpan data identitas yang digunakan untuk admin saat ingin *login*. Tabel ini memiliki 4 *fields* antara lain *id*, *email*, *name*, dan *password*. Struktur tabel kecamatan disajikan dalam tabel 4.16.

Tabel 4.15 Struktur Tabel Kecamatan

No	Nama <i>Field</i>	Tipe	Keterangan
1	id	<i>int(10)</i>	<i>Primary key</i>
2	id_kab	<i>int(11)</i>	<i>Foreign key</i> yang me-relasi kan tabel ini dengan tabel kabupaten
3	nama_kecamatan	<i>varchar(255)</i>	Nama kecamatan

Tabel 4.16 Struktur Tabel *User*

No	Nama <i>Field</i>	Tipe	Keterangan
1	id	<i>int(10)</i>	<i>Primary key</i>
2	<i>name</i>	<i>varchar(255)</i>	<i>Field</i> untuk menyimpan <i>username</i> admin. <i>Field</i> ini adalah <i>field</i> yang di-generate oleh <i>framework Laravel</i>
3	<i>email</i>	<i>varchar(255)</i>	<i>Field</i> untuk menyimpan <i>email</i> admin. <i>Field</i> ini adalah <i>field</i> yang di-generate oleh <i>framework Laravel</i>

Tabel 4.17 Struktur Tabel *User* (Lanjutan)

No	Nama <i>Field</i>	Tipe	Keterangan
4	<i>password</i>	<i>varchar(255)</i>	<i>Field</i> untuk menyimpan <i>password</i> admin yang telah dienkripsi. <i>Field</i> ini adalah <i>field</i> yang di-generate oleh <i>framework Laravel</i>

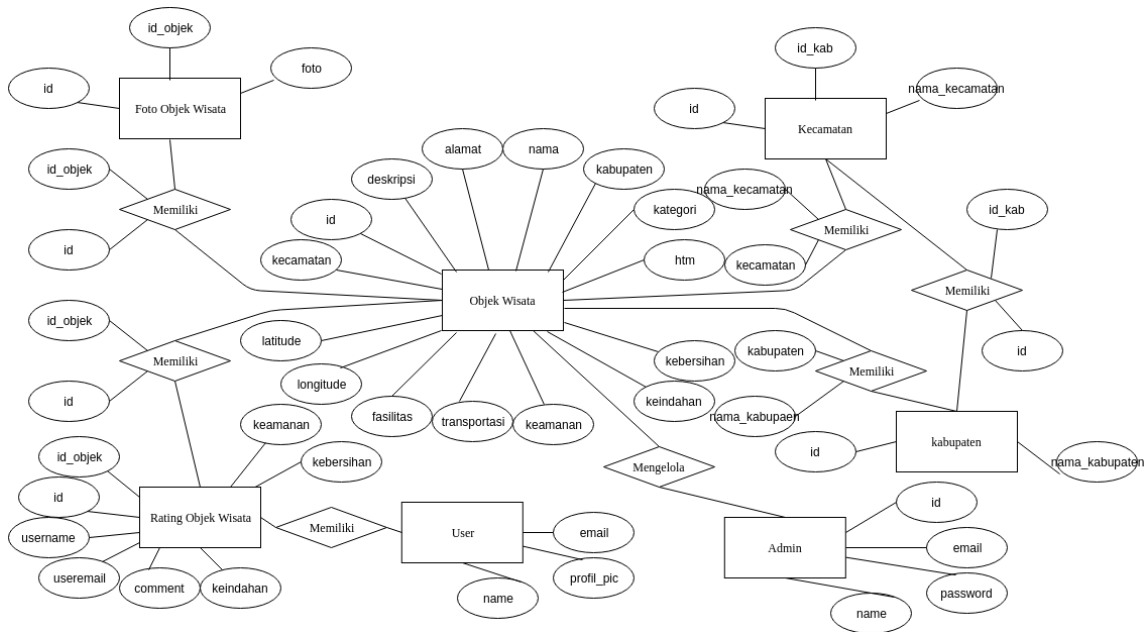
4.2.6. Rancangan ERD(*Entity Relationship Diagram*)

Dalam perancangan ERD terdapat hubungan antara tabel satu dengan yang lain.

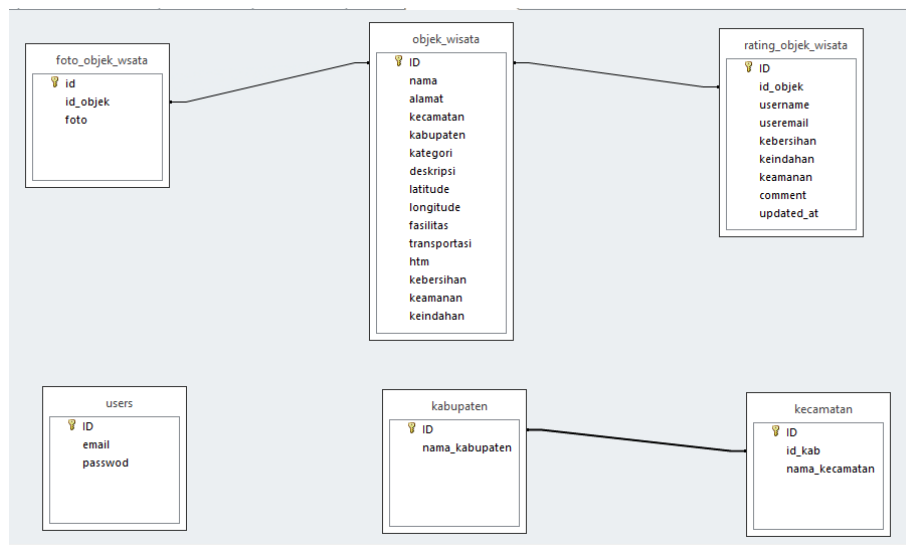
Penjelasan relasi antara tabel pada sistem ini adalah sebagai berikut :

1. Objek wisata memiliki kecamatan, kabupaten, foto objek wisata dan rating objek wisata
2. Kabupaten memiliki kecamatan
3. *User* dapat menambah rating objek wisata
4. Admin mengelola objek wisata dan foto objek wisata

Rancangan ERD sistem ini disajikan pada Gambar 4.26 dan relasi tabel *database* disajikan pada Gambar 2.7.



Gambar 4.26 Rancangan ERD Sistem



Gambar 4.27 Rancangan Relasi Tabel Database

4.2.7. Perancangan Antar Muka Sistem

Dalam pembangunan sistem, diperlukan perancangan antarmuka untuk memudahkan pengguna dalam melakukan proses interaksi terhadap sistem. Antarmuka menyediakan halaman sebuah sistem yang digunakan untuk proses masukan data hingga menghasilkan antarmuka keluaran yang sesuai dengan kebutuhan. Rancangan antarmuka

untuk sistem yang akan dikembangkan adalah antara lain:

1. **Rancangan Antarmuka *Login Admin***

Rancangan Antarmuka *Login Admin* adalah halaman yang ditampilkan saat admin ingin mengakses halaman admin. Halaman ini akan menampilkan dua *input text* untuk memasukkan *username* dan *password* serta satu *button* untuk melakukan *submit* data. Rancangan antarmuka halaman ini terlihat seperti Gambar 4.28.

2. **Rancangan Antarmuka *Home Admin***

Rancangan Antarmuka *Login Admin* adalah halaman yang ditampilkan saat admin ingin mengakses halaman admin. Halaman ini akan menampilkan dua *input text* untuk memasukkan *username* dan *password* serta satu *button* untuk melakukan *submit* data. Rancangan antarmuka halaman ini terlihat seperti Gambar 4.29.

3. **Rancangan Antarmuka *Daftar Objek Wisata Admin***

Rancangan Antarmuka ini adalah halaman yang ditampilkan saat admin ingin melihat daftar objek wisata apa saja yang ada di database. Halaman ini akan menampilkan tabel dimana berisi sedikit informasi tentang objek wisata dan dua tombol untuk mend-*edit* dan menghapus data objek wisata. Rancangan antarmuka halaman ini terlihat seperti Gambar 4.30.





 SELAMAT DATANG DI EXPLORE JOGJA

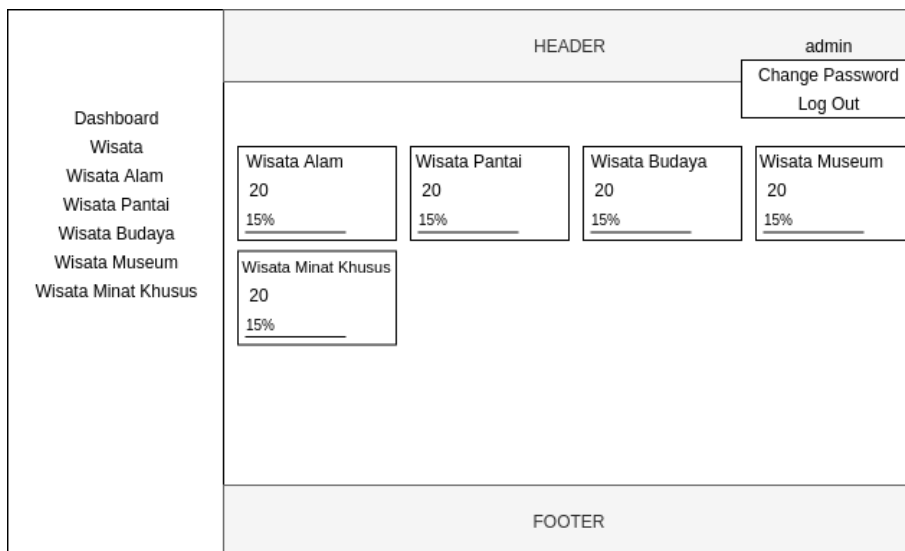
Silahkan masukkan username dan password

Username

Password

Login

Gambar 4.28 Rancangan Antarmuka Login Admin



HEADER					
Dashboard Wisata Wisata Alam Wisata Pantai Wisata Budaya Wisata Museum Wisata Minat Khusus	admin			Change Password	
	Log Out				
	Wisata Alam 20 15%	Wisata Pantai 20 15%	Wisata Budaya 20 15%	Wisata Museum 20 15%	
	Wisata Minat Khusus 20 15%				
FOOTER					

Gambar 4.29 Rancangan Antarmuka Home Admin

HEADER						
admin						
Dashboard Wisata Wisata Alam Wisata Pantai Wisata Budaya Wisata Museum Wisata Minat Khusus	Data Objek Wisata					
	<input type="button" value="Tambah Wisata"/>					
	<input type="button" value="Cari Wisata"/> <input type="text"/> <input type="text"/> <input type="text"/>					
	No	Nama	Lokasi	Kategori	Rating	Aksi
	1					
	2					
	3					
	4					
	5					
	6					
7						
8						
9						
10						
1 2 3 4 >						
FOOTER						

Gambar 4.30 Rancangan Antarmuka Daftar Objek Wisata Admin

4. Rancangan Antarmuka Tambah Objek Wisata Admin

Rancangan Antarmuka ini adalah halaman yang ditampilkan saat admin ingin menambahkan objek wisata ke database. Halaman ini akan menampilkan *form-form* yang harus diisi oleh admin tentang detail objek wisata. Rancangan antarmuka halaman ini terlihat seperti Gambar 4.31.

HEADER		admin
Dashboard Wisata Wisata Alam Wisata Pantai Wisata Budaya Wisata Museum Wisata Minat Khusus	Tambah Objek Wisata	
	Kategori Objek Wisata	Fasilitas <input type="checkbox"/> Mushola <input type="checkbox"/> Toilet <input type="checkbox"/> Tempat Parkir <input type="checkbox"/> Restoran <input type="checkbox"/> Penginapan
	Nama Objek Wisata	Transportasi
	Kabupaten	Harga Tiket Masuk
	Kecamatan	
	Alamat	
	Lokasi	
	Maps	
	Latitude	Longitude
	Deskripsi Deskripsi	Foto <input type="button" value="Choose File"/> <input type="button" value="Tambah Foto"/>
<input type="button" value="SUBMIT"/>		
FOOTER		

Gambar 4.31 Rancangan Antarmuka Tambah Objek Wisata Admin

5. Rancangan Antarmuka *Edit* Objek Wisata Admin

Rancangan Antarmuka ini hampir sama dengan rancangan antarmuka tambah objek wisata. Halaman ini memuat *form-form* yang harus diisi oleh admin tentang detail objek wisata. Namun, *form-form* tadi telah diisi oleh data objek wisata yang telah ada. Rancangan antarmuka halaman ini terlihat seperti Gambar 4.32.

6. Rancangan Antarmuka Mengubah *Password* Admin

Rancangan Antarmuka ini memuat *form* untuk mengubah *password* untuk *login*

admin. Rancangan antarmuka halaman ini terlihat seperti Gambar 4.33.

7. Rancangan Antarmuka *Splash Screen User*

Rancangan Antarmuka ini *splash screen* sebelum *user* masuk ke halaman utama aplikasi *mobile*. Rancangan antarmuka halaman ini terlihat seperti Gambar 4.34.

8. Rancangan Antarmuka *Login User*

Rancangan Antarmuka ini muncul saat *splash screen*, *user* diharuskan memilih salah satu *google account* yang terdaftar pada *smartphone* mereka. Rancangan antarmuka halaman ini terlihat seperti Gambar 4.35.

9. Rancangan Antarmuka *Home User*

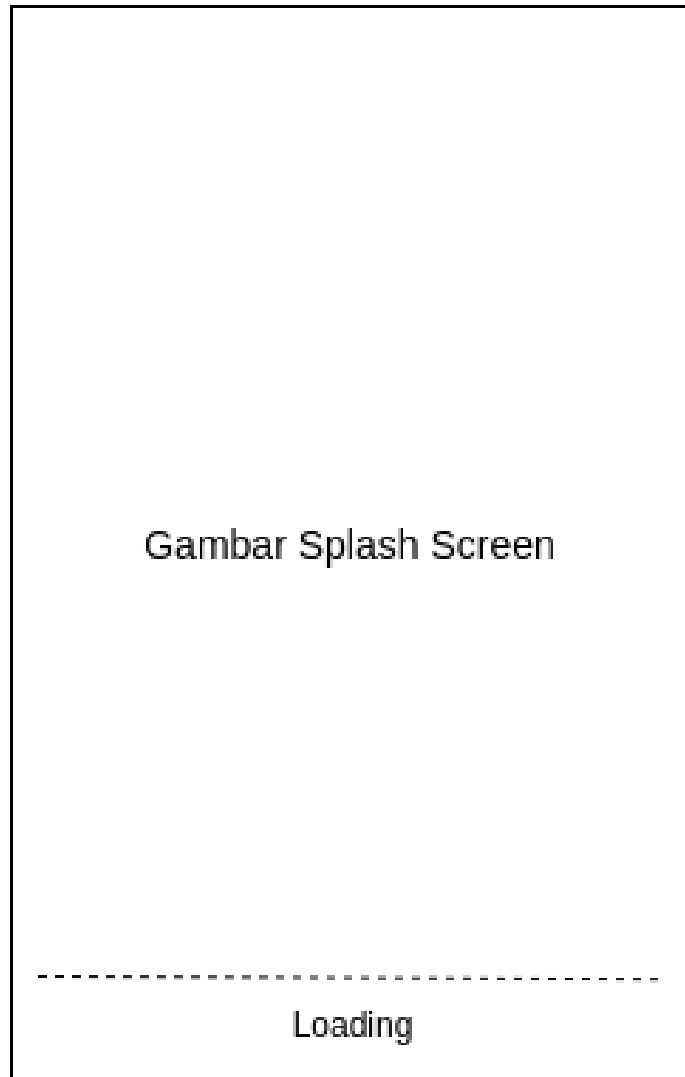
Rancangan Antarmuka ini merupakan tampilan utama bagi *user*. Dalam rancangan antarmuka ini, menampilkan daftar objek wisata yang ada dalam database. Di dalamnya juga termuat sebuah *text input* untuk mencari objek wisata yang ingin *user* lihat dan sebuah tombol untuk memunculkan menu memasukkan kriteria yang diinginkan *user*. Juga ditampilkan gambar profil *google account* yang digunakan *user* pada aplikasi ini. Rancangan antarmuka halaman ini terlihat seperti Gambar 4.36.

HEADER		admin
Dashboard Wisata Wisata Alam Wisata Pantai Wisata Budaya Wisata Museum Wisata Minat Khusus	Edit Objek Wisata	
	Kategori Objek Wisata	Fasilitas <input type="checkbox"/> Mushola <input type="checkbox"/> Toilet <input type="checkbox"/> Tempat Parkir <input type="checkbox"/> Restoran <input type="checkbox"/> Penginapan
	Nama Objek Wisata	Transportasi
	Kabupaten	Harga Tiket Masuk
	Kecamatan	
	Alamat	
	Lokasi	
	Maps	
	Latitude	Longitude
	Deskripsi Deskripsi	Foto <input type="button" value="Choose File"/> <input type="button" value="Tambah Foto"/>
Foto-foto Objek Wisata		
No	Gambar	Aksi
<input type="button" value="SUBMIT"/>		
FOOTER		

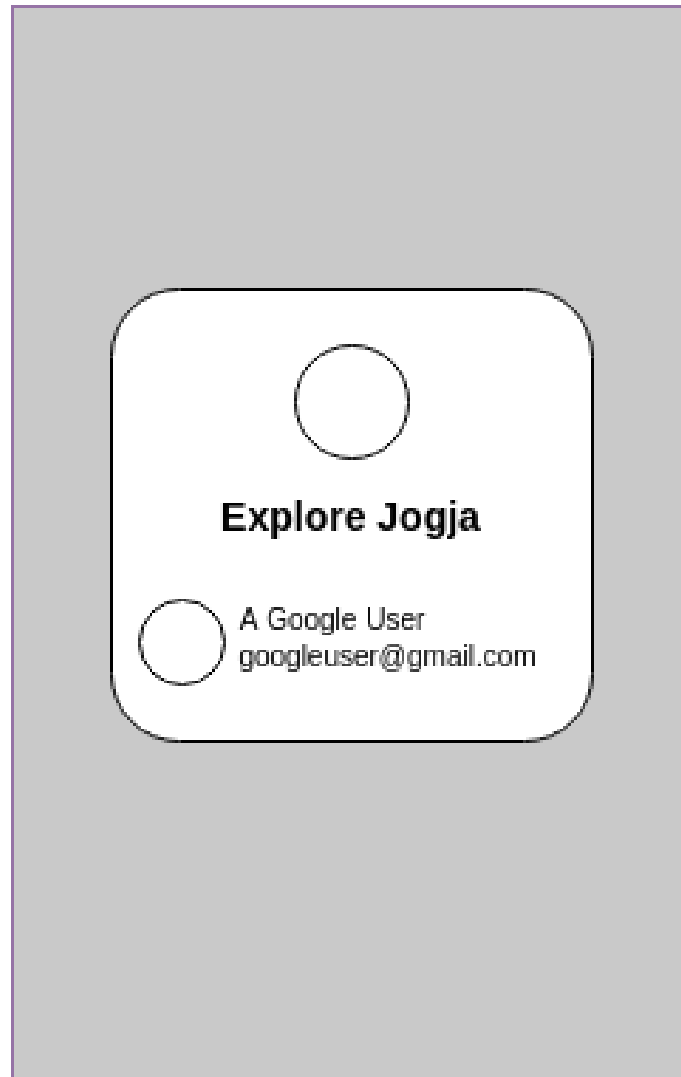
Gambar 4.32 Rancangan Antarmuka *Edit* Objek Wisata Admin

Dashboard Wisata Wisata Alam Wisata Pantai Wisata Budaya Wisata Museum Wisata Minat Khusus	HEADER	admin
	Ganti Password	
	<input type="text" value="Password Lama"/>	
	Password Baru	
<input type="text" value="Password Baru"/>		
<input type="text" value="Konfirmasi Password Baru"/>		
<input type="button" value="SUBMIT"/>		
FOOTER		

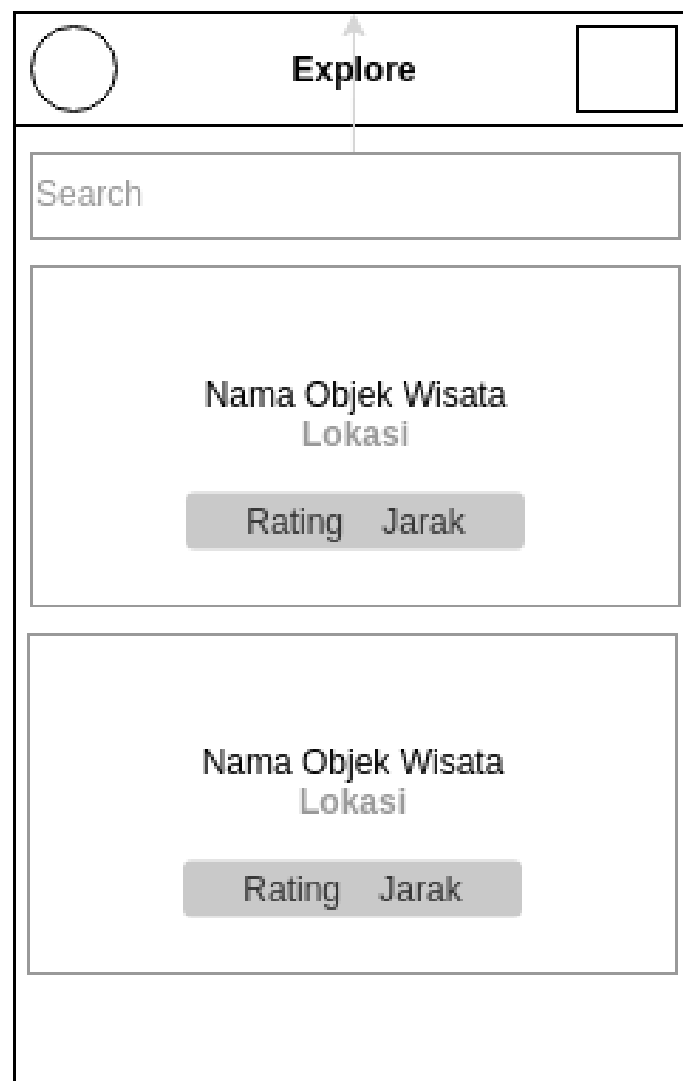
Gambar 4.33 Rancangan Antarmuka Mengubah *Password Admin*



Gambar 4.34 Rancangan Antarmuka *Splash Screen User*



Gambar 4.35 Rancangan Antarmuka *Login User*



Gambar 4.36 Rancangan Antarmuka *Home User*

10. Rancangan Antarmuka Masukkan Kriteria

Rancangan Antarmuka ini muncul saat *user* menekan tombol masukkan kriteria. *User* diharuskan memasukkan seberapa penting 7 kriteria saat akan memilih objek wisata dalam skala 1 sampai 5. Rancangan antarmuka halaman ini terlihat seperti Gambar 4.37.

11. Rancangan Antarmuka *Log Out User*

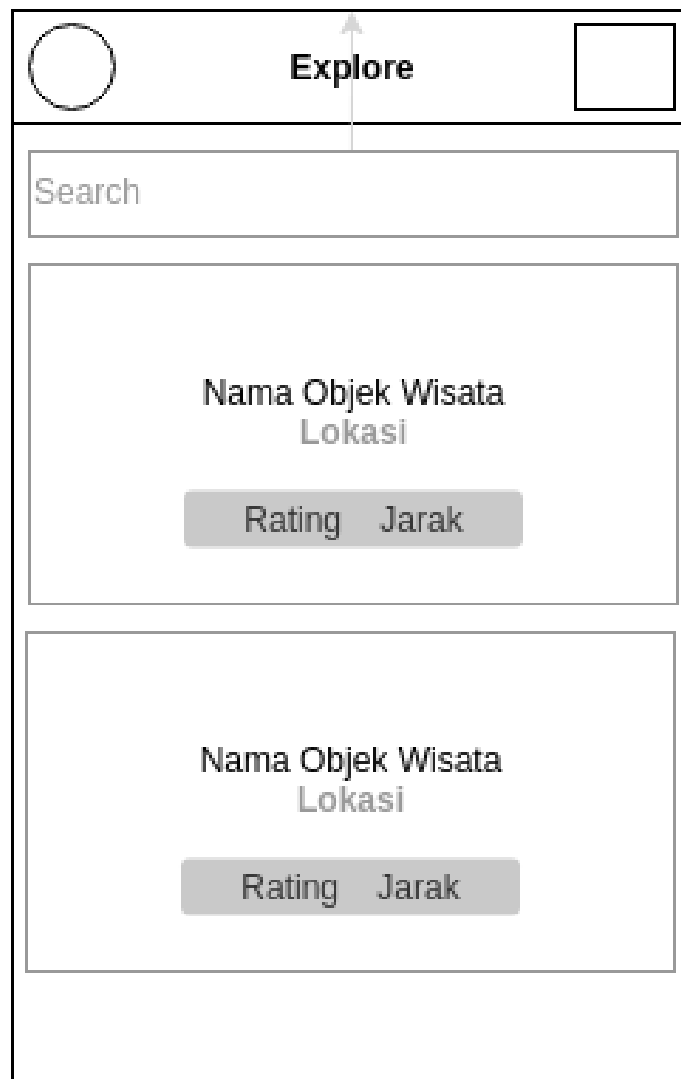
Rancangan Antarmuka ini muncul saat *user* menekan gambar profil pada menu

utama. *User log out* dan harus memilih ulang satu *google account* yang terdaftar pada *smartphone*. Rancangan antarmuka halaman ini terlihat seperti Gambar 4.38.

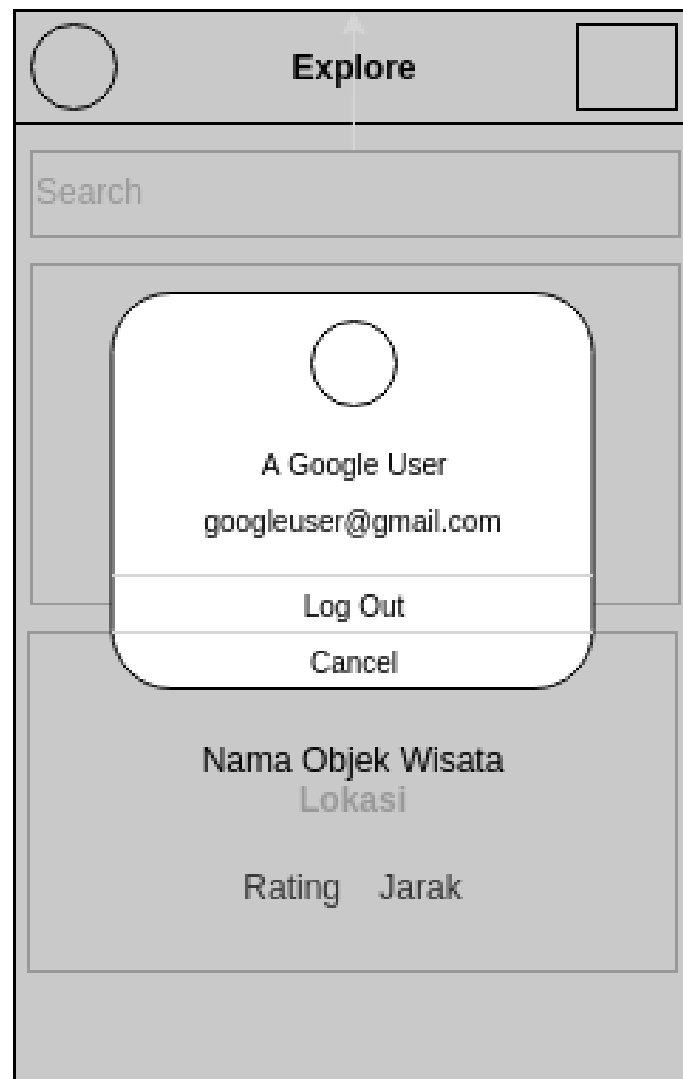
12. Rancangan Antarmuka Detail Objek Wisata

Rancangan Antarmuka ini muncul saat *user* memilih salah satu objek wisata. Rancangan antarmuka ini menampilkan seluruh detail dari objek wisata yang dipilih.

Rancangan antarmuka halaman ini terlihat seperti Gambar 4.39.



Gambar 4.37 Rancangan Antarmuka *Home User*



Gambar 4.38 Rancangan Antarmuka *Log Out User*



Gambar 4.39 Rancangan Antarmuka Detail Objek Wisata

13. **Rancangan Antarmuka *Edit* atau Tambah Penilaian Objek Wisata**

Rancangan Antarmuka ini muncul saat *user* akan menambahkan atau meng-*edit* penilaian mereka terhadap sebuah objek wisata. Ada beberapa form yang harus diisi oleh *user*, diantaranya tingkat kebersihan, keamanan dan keindahan serta komentar mereka terhadap objek wisata tersebut. Rancangan antarmuka halaman ini terlihat seperti Gambar 4.40.

14. **Rancangan Antarmuka Lihat Seluruh Komentar Objek Wisata**

Rancangan Antarmuka ini menampilkan seluruh komentar dari *user* terhadap suatu objek wisata. Rancangan antarmuka halaman ini terlihat seperti Gambar 4.41.

15. **Rancangan Antarmuka Lihat Gambar Galeri Objek Wisata**

Rancangan Antarmuka ini menampilkan gambar galeri sebuah objek wisata secara *fullscreen*. Rancangan antarmuka halaman ini terlihat seperti Gambar 4.42.

Nama Objek Wisata

Lo

G

Fa

D

○

A Google User

View ★★★★★

Clean ★★★★★

Security ★★★★★

Comment

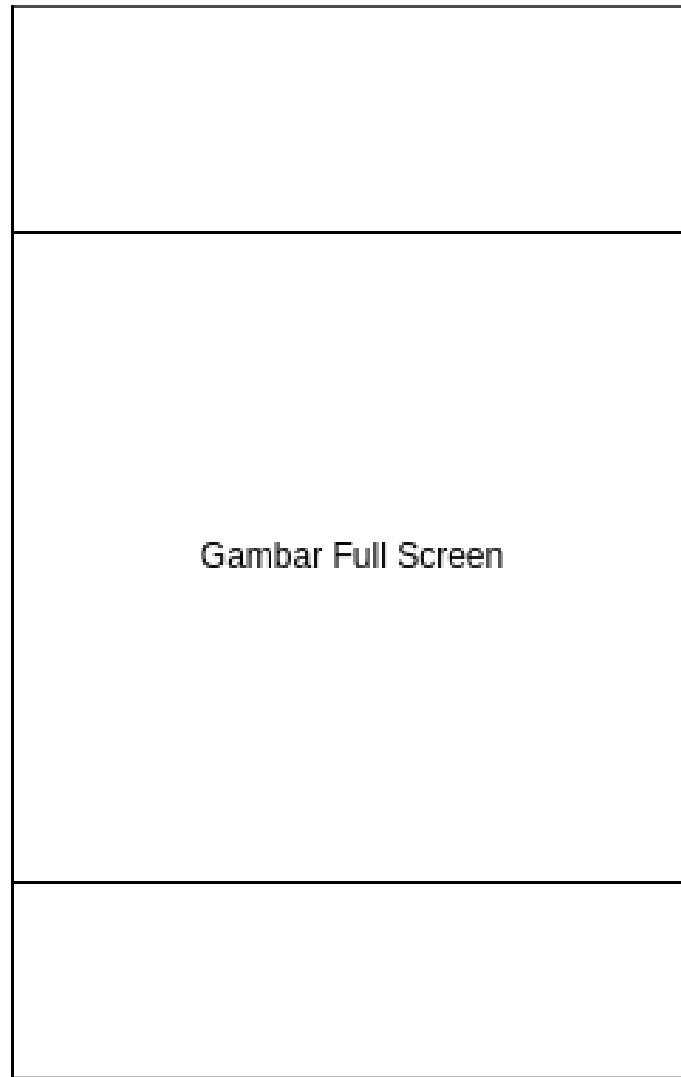
Submit

Delete

Gambar 4.40 Rancangan Antarmuka *Edit* atau Tambah Penilaian Objek Wisata

Nama Objek Wisata
Jumlah Review
A Google User 12/03/19
Bagus Tempatnya
A Google User 12/03/19
Bagus Tempatnya
A Google User 12/03/19
Bagus Tempatnya
A Google User 12/03/19
Bagus Tempatnya

Gambar 4.41 Rancangan Antarmuka Lihat Seluruh Komentar Objek Wisata



Gambar 4.42 Rancangan Antarmuka Lihat Gambar Galeri Objek Wisata

BAB V

IMPLEMENTASI DAN PENGUJIAN SISTEM

5.1. Implementasi

Implementasi merupakan tahap penerjemahan hasil perancangan aplikasi sistem pendukung keputusan yang telah dilakukan sebelumnya dalam bentuk baris-baris kode program. Dalam implementasinya peneliti menggunakan bahasa pemrograman PHP, html, css dan database MySQL serta *laravel framework* sebagai media penyimpanan data. Sementara untuk menampilkan data kepada *user* menggunakan bahasa pemrograman Kotlin.

5.1.1. Implementasi *Database*

Implementasi database pada sistem pendukung keputusan ini menggunakan database MySQL. Alasan menggunakan database MySQL adalah, karena mempunyai desain interface yang mudah digunakan, sintaksnya lebih mudah dipahami dan tidak terlalu rumit, dapat diperoleh secara gratis, dan memiliki kemampuan yang cukup handal. Implementasi database pada sistem disajikan dalam Gambar 5.1 dan relasi database disajikan dalam Gambar 5.2.

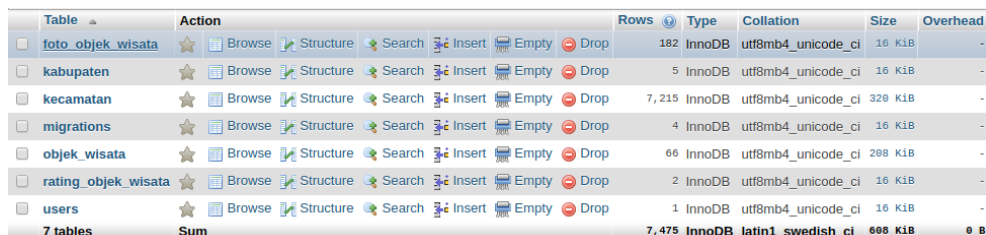
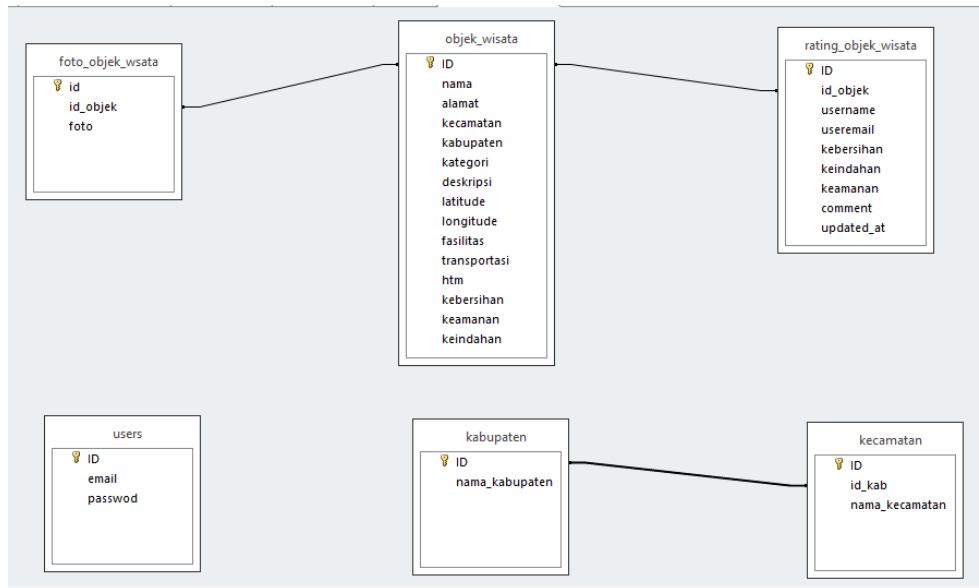


Table	Action	Rows	Type	Collation	Size	Overhead
foto_objek_wisata	Browse Structure Search Insert Empty Drop	182	InnoDB	utf8mb4_unicode_ci	16 KiB	-
kabupaten	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_unicode_ci	16 KiB	-
kecamatan	Browse Structure Search Insert Empty Drop	7,215	InnoDB	utf8mb4_unicode_ci	320 KiB	-
migrations	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_unicode_ci	16 KiB	-
objek_wisata	Browse Structure Search Insert Empty Drop	66	InnoDB	utf8mb4_unicode_ci	208 KiB	-
rating_objek_wisata	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_unicode_ci	16 KiB	-
users	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_unicode_ci	16 KiB	-
7 tables	Sum	7,475	InnoDB	latin1_swedish_ci	608 KiB	0 B

Gambar 5.1 Implementasi *Database*



Gambar 5.2 Relasi Database

5.1.2. Implementasi Halaman Sistem

1. Halaman *Login Admin*

Halaman yang ditampilkan saat admin ingin mengakses halaman admin. Halaman ini akan menampilkan dua *input text* untuk memasukkan *username* dan *password* serta satu *button* untuk melakukan *submit* data. Halaman antarmuka ini terlihat seperti pada Gambar 5.3.

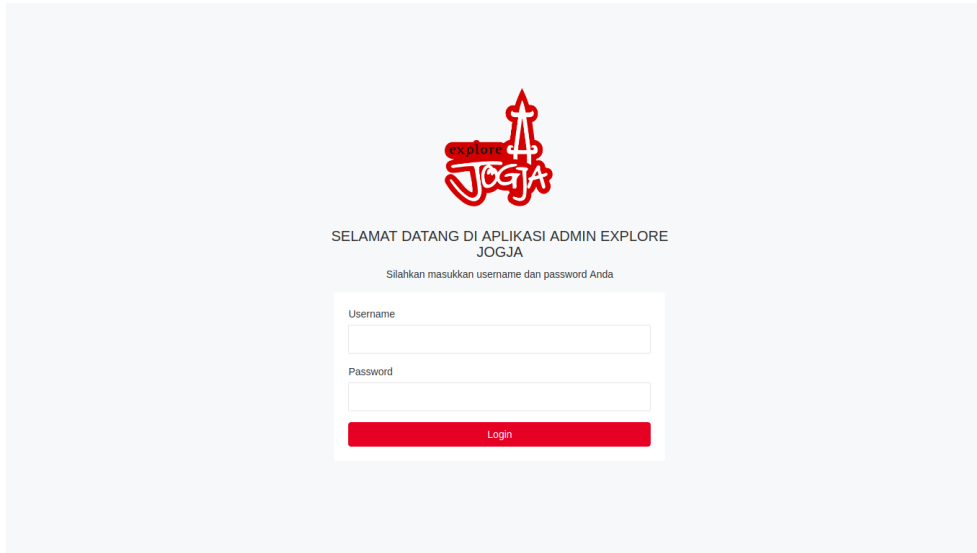
2. Halaman *Home Admin*

Halaman yang ditampilkan saat admin berhasil *login*. Halaman ini menampilkan statistik jumlah objk wisata yang ada. Halaman antarmuka ini terlihat seperti pada Gambar 5.4.

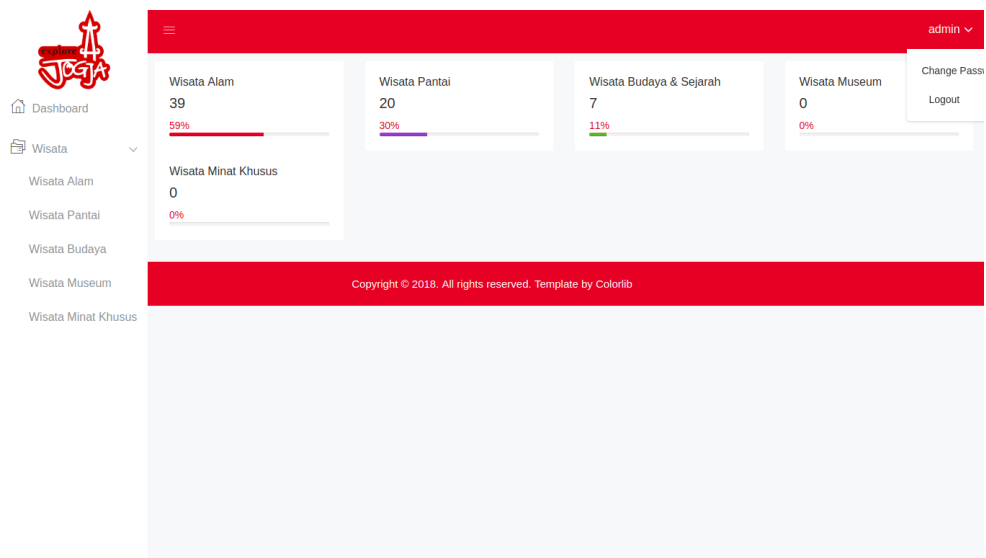
3. Halaman *Daftar Objek Wisata Admin*

Halaman yang ditampilkan saat admin ingin melihat daftar objek wisata apa saja yang ada di database. Halaman ini akan menampilkan tabel dimana berisi sedikit

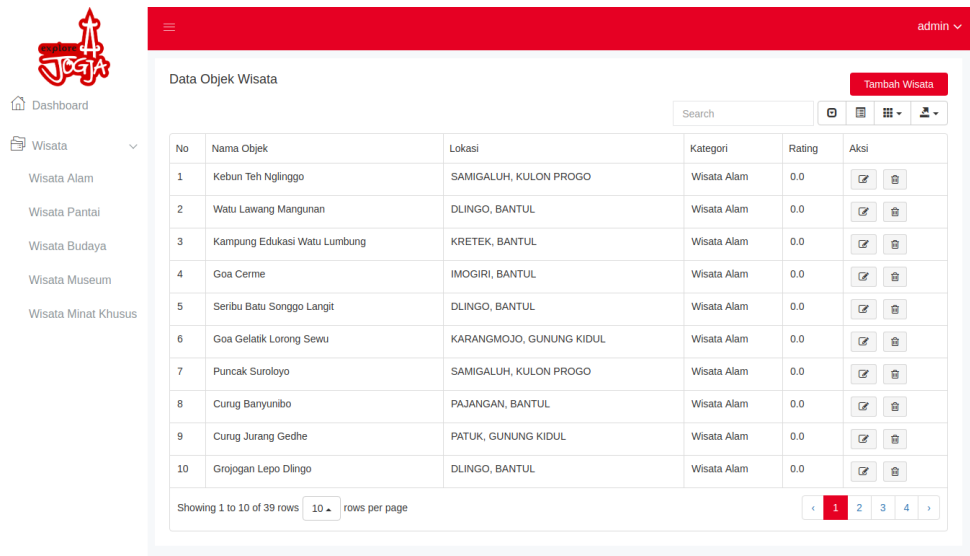
informasi tentang objek wisata dan dua tombol untuk meng-*edit* dan menghapus data objek wisata. Halaman antarmuka ini terlihat seperti pada Gambar 5.5.



Gambar 5.3 Halaman *Login* Admin



Gambar 5.4 Halaman *Home* Admin



The screenshot shows an admin interface for managing tourism objects. The main content area is titled 'Data Objek Wisata' and contains a table with the following data:

No	Nama Objek	Lokasi	Kategori	Rating	Aksi
1	Kebun Teh Nglingsgo	SAMIGALUH, KULON PROGO	Wisata Alam	0.0	[Edit] [Delete]
2	Watu Lawang Mangunan	DLINGO, BANTUL	Wisata Alam	0.0	[Edit] [Delete]
3	Kampung Edukasi Watu Lumbung	KRETEK, BANTUL	Wisata Alam	0.0	[Edit] [Delete]
4	Goa Cerme	IMOGIRI, BANTUL	Wisata Alam	0.0	[Edit] [Delete]
5	Seribu Batu Songgo Langit	DLINGO, BANTUL	Wisata Alam	0.0	[Edit] [Delete]
6	Goa Gelatik Lorong Sewu	KARANGMOJO, GUNUNG KIDUL	Wisata Alam	0.0	[Edit] [Delete]
7	Puncak Suroloyo	SAMIGALUH, KULON PROGO	Wisata Alam	0.0	[Edit] [Delete]
8	Curug Banyunibo	PAJANGAN, BANTUL	Wisata Alam	0.0	[Edit] [Delete]
9	Curug Jurang Gedhe	PATUK, GUNUNG KIDUL	Wisata Alam	0.0	[Edit] [Delete]
10	Grojogan Lepo Dlingo	DLINGO, BANTUL	Wisata Alam	0.0	[Edit] [Delete]

Below the table, there is a pagination control showing 'Showing 1 to 10 of 39 rows' and '10 rows per page'. The page number '1' is highlighted in red.

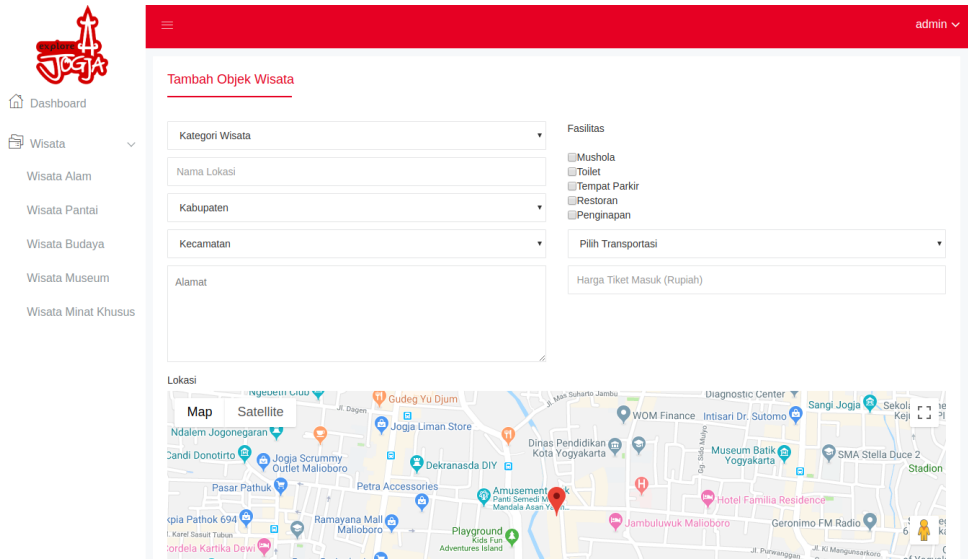
Gambar 5.5 Halaman Daftar Objek Wisata Admin

4. Halaman Tambah Objek Wisata Admin

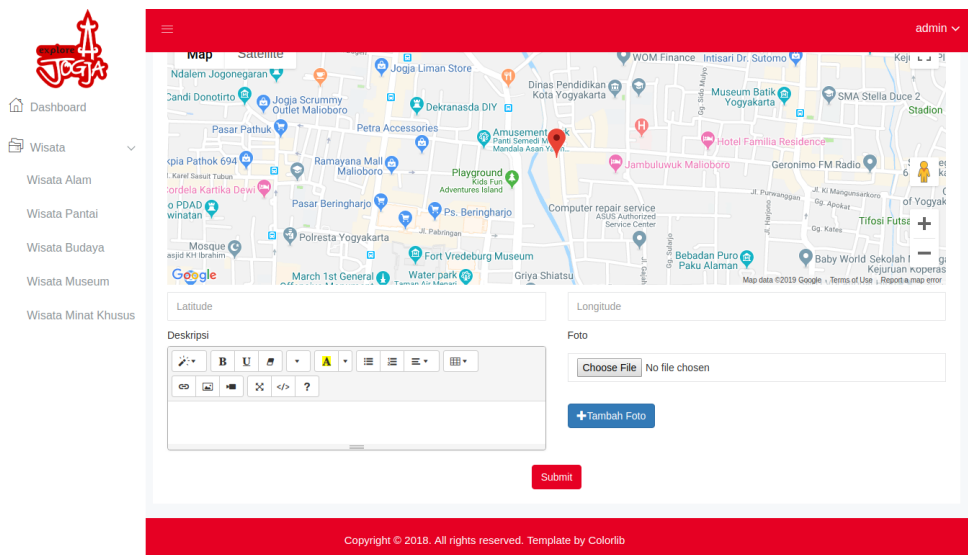
Halaman yang ditampilkan saat admin ingin menambah objek wisata ke database. Halaman ini akan menampilkan *form-form* yang harus diisi oleh admin tentang detail objek wisata. Halaman antarmuka ini terlihat seperti pada Gambar 5.6.

5. Halaman *Edit* Objek Wisata Admin

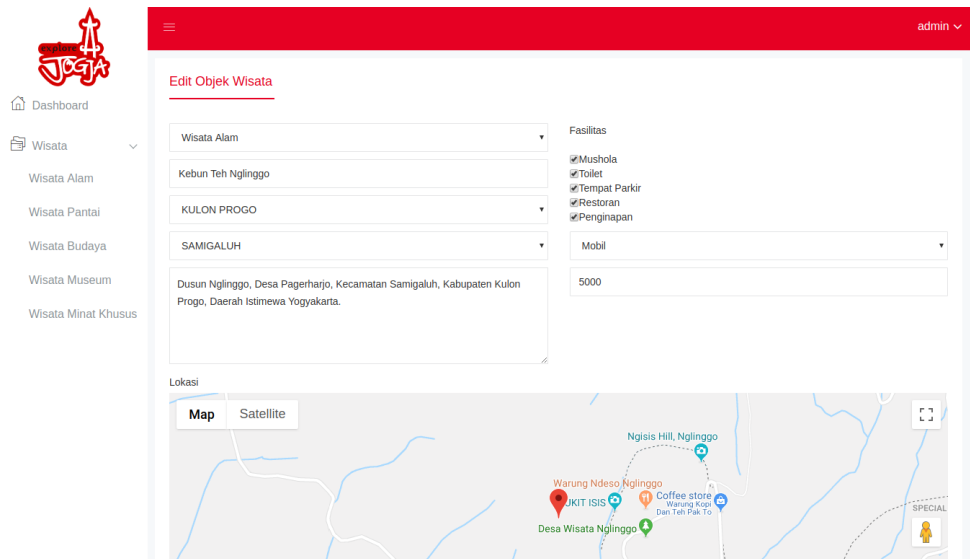
Halaman ini hampir sama dengan halaman tambah objek wisata. Halaman ini memuat *form-form* yang harus diisi oleh admin tentang detail objek wisata. Namun, *form-form* tadi telah diisi oleh data objek wisata yang telah ada. Halaman antarmuka ini terlihat seperti pada Gambar 5.8.



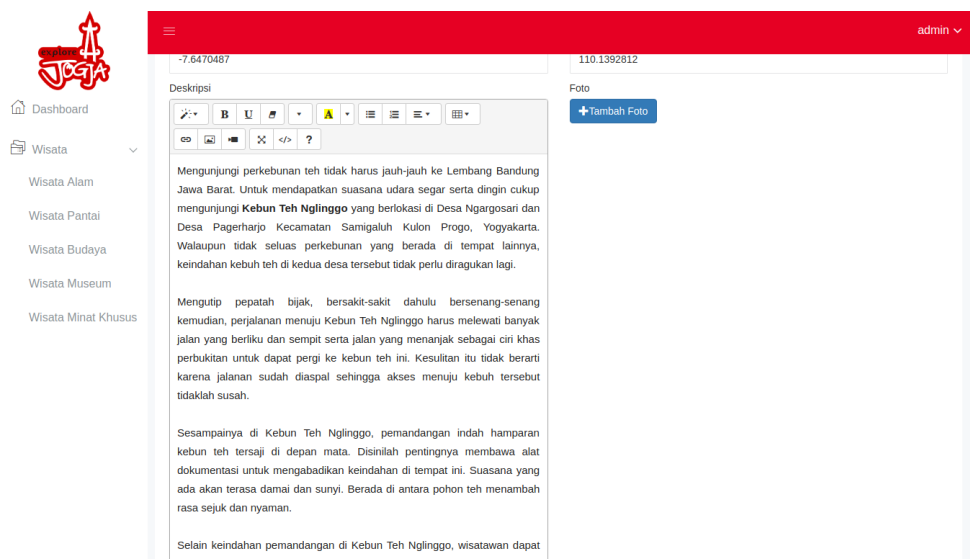
Gambar 5.6 Halaman Tambah Objek Wosata Admin



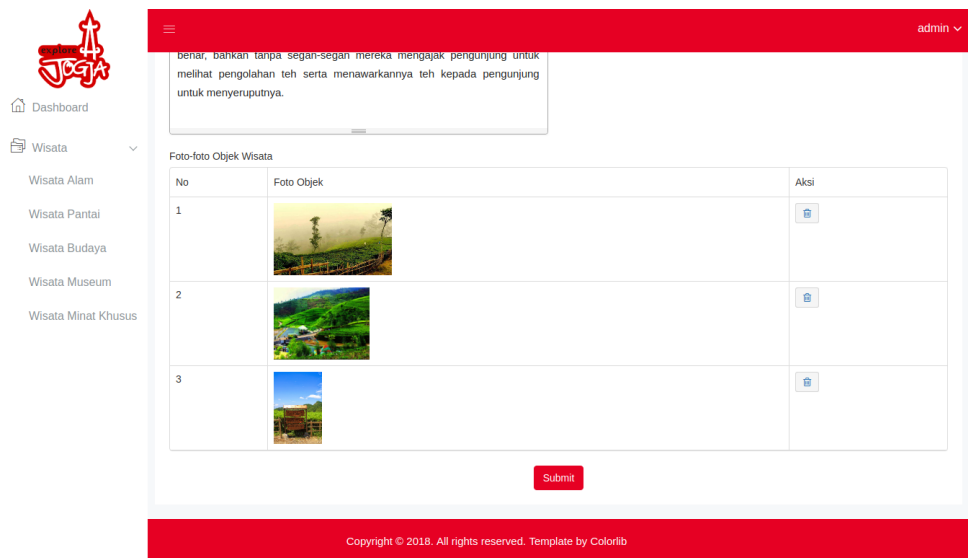
Gambar 5.7 Halaman Tambah Objek Wisata Admin (Lanjutan)



Gambar 5.8 Halaman *Edit* Objek Wisata Admin



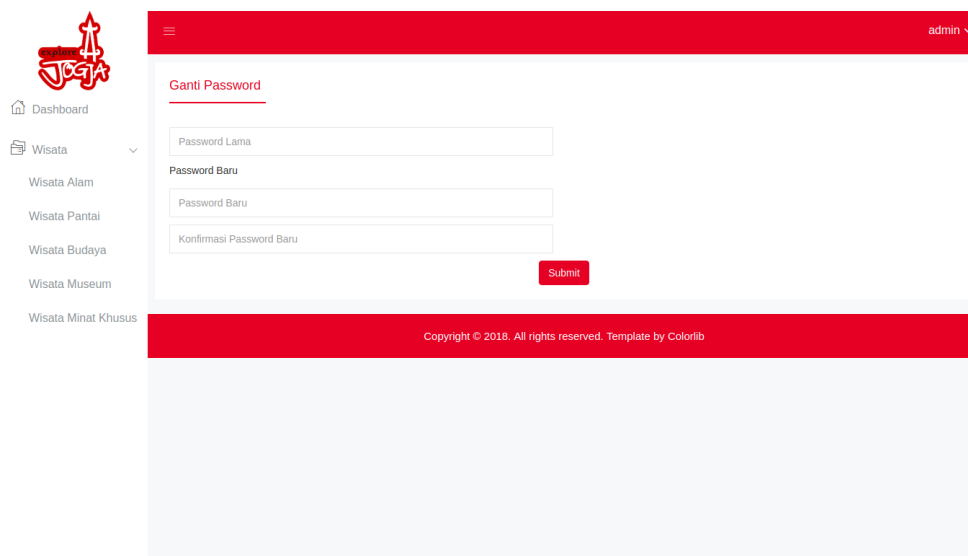
Gambar 5.9 Halaman *Edit* Objek Wisata Admin (Lanjutan)



Gambar 5.10 Halaman *Edit* Objek Wisata Admin (Lanjutan)

6. Halaman Mengubah *Password* Admin

Halaman ini memuat *form* untuk mengubah *password* untuk *login* admin. Halaman antarmuka ini terlihat seperti pada Gambar 5.11.



Gambar 5.11 Halaman Mengubah *Password* Admin

7. Halaman *Splash Screen* User

Halaman ini *splash screen* sebelum user masuk ke halaman utama aplikasi *mobile*.

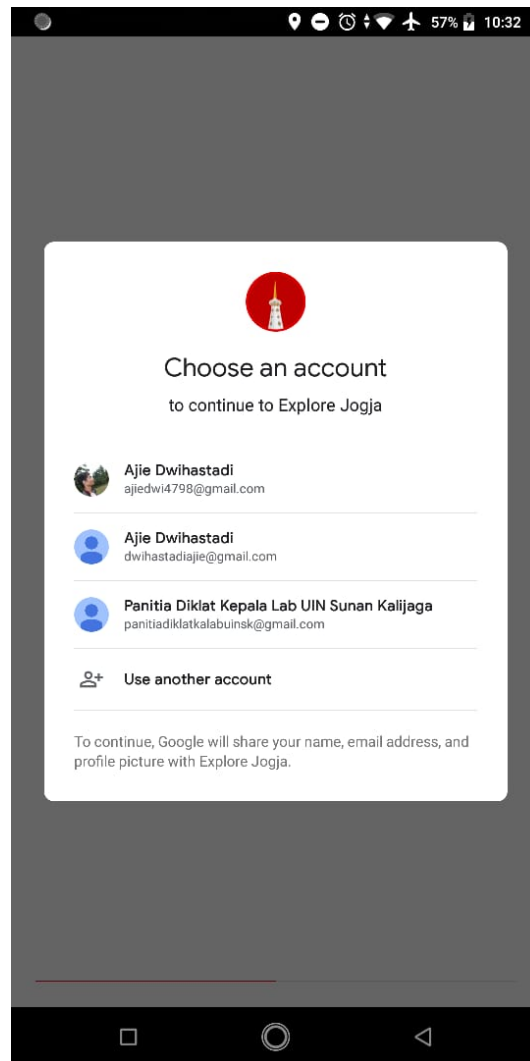
Halaman antarmuka ini terlihat seperti pada Gambar 5.12.

8. Halaman *Login User*

Halaman ini muncul saat *splash screen*, *user* diharuskan memilih salah satu *google account* yang terdaftar pada *smartphone* mereka. Halaman antarmuka ini terlihat seperti pada Gambar 5.13.



Gambar 5.12 Halaman *Splash Screen User*



Gambar 5.13 Halaman *Login User*

9. Halaman *Home User*

Halaman ini merupakan tampilan utama bagi *user*. Dalam halaman ini, menampilkan daftar objek wisata yang ada dalam database. Di dalam nya juga termuat sebuah *text input* untuk mencari objek wisata yang ingin *user* lihat dan sebuah tombol untuk memunculkan menu memasukkan kriteria yang diinginkan *user*. Juga ditampilkan gambar profil *google account* yang digunakan *user* pada aplikasi ini. Halaman antarmuka ini terlihat seperti pada Gambar 5.14.

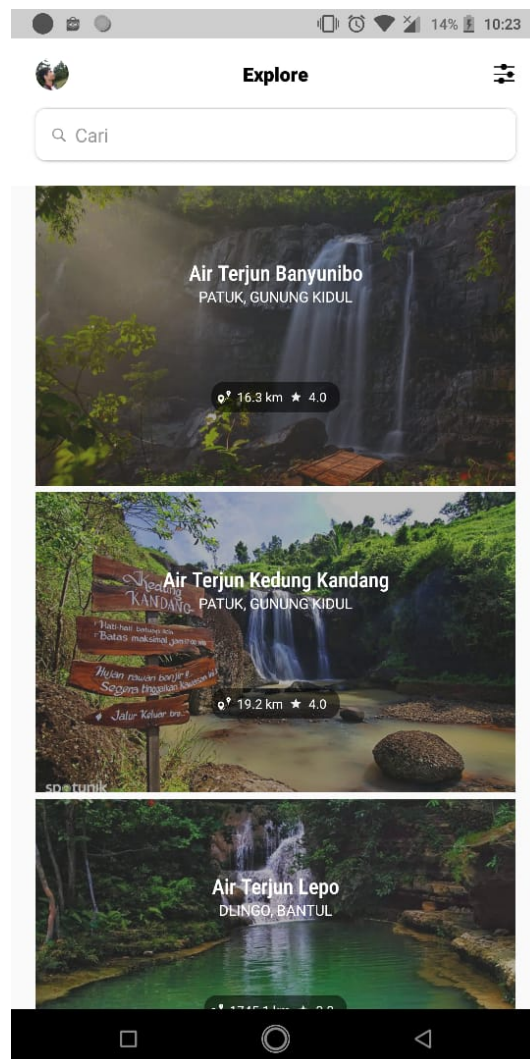
10. Halaman Masukkan Kriteria

Halmaan ini muncul saat *user* menekan tombol masukkan kriteria. *User* diharuskan memasukkan seberapa penting 7 kriteria saat akan memilih objek wisata dalam skala 1 sampai 5. Halaman antarmuka ini terlihat seperti pada Gambar 5.15.

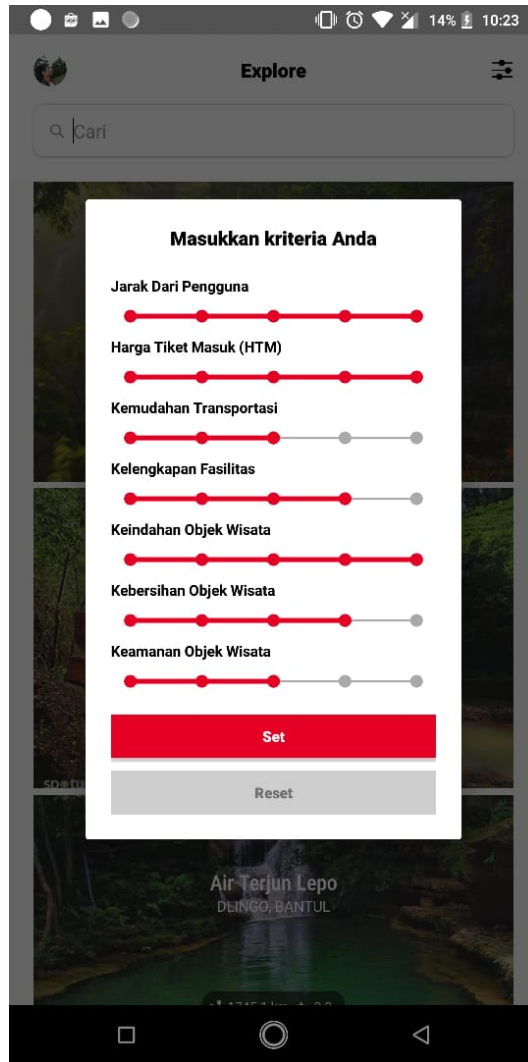
11. Halaman *Log Out User*

Halaman ini muncul saat *user* menekan gambar profil pada menu utama. *User log out* dan harus memilih ulang satu *google account* yang terdaftar pada *smartphone*.

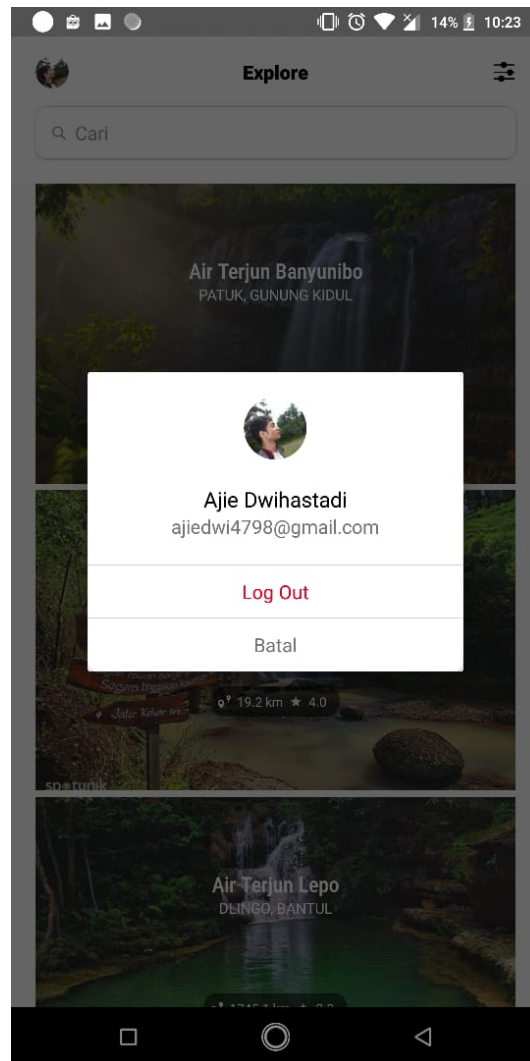
Halaman antarmuka ini terlihat seperti pada Gambar 5.16.



Gambar 5.14 Halaman *Home User*



Gambar 5.15 Halaman Masukkan Kriteria



Gambar 5.16 Halaman *Log Out User*

12. Halaman Detail Objek Wisata

Halaman ini muncul saat *user* memilih salah satu objek wisata. Halaman ini menampilkan seluruh detail dari objek wisata yang dipilih. Halaman antarmuka ini terlihat seperti pada Gambar 5.17.

13. Halaman *Edit* atau Tambah Penilaian Objek Wisata

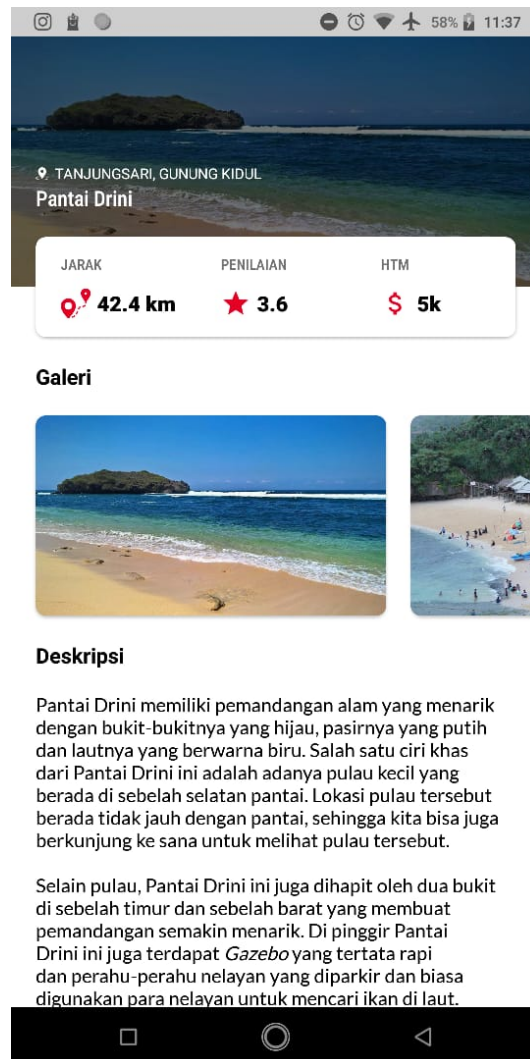
Halaman ini muncul saat *user* akan menambahkan atau meng-*edit* penilaian mereka terhadap sebuah objek wisata. Ada beberapa *form* yang harus diisi oleh *user*, diantaranya tingkat kebersiha, keamanan dan keindahan serta komentar mereka terhadap objek wisata tersebut. Halaman antarmuka ini terlihat seperti pada Gambar 5.19.

14. Halaman Lihat Seluruh Komentar Objek Wisata

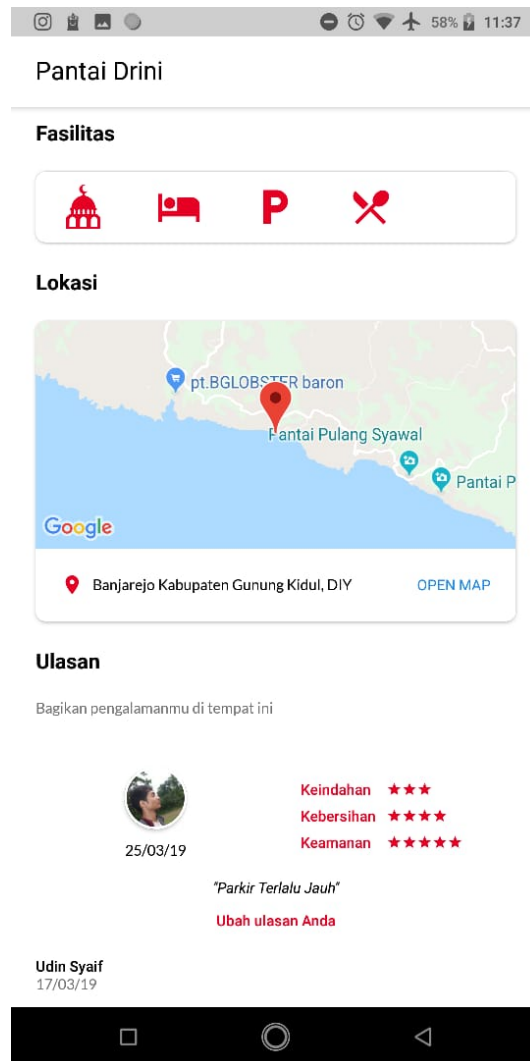
Halaman ini menampilkan seluruh komentar dari *user* terhadap suatu objek wisata. Halaman antarmuka ini terlihat seperti pada Gambar 5.20.

15. Halaman Lihat Gambar Galeri Objek Wisata

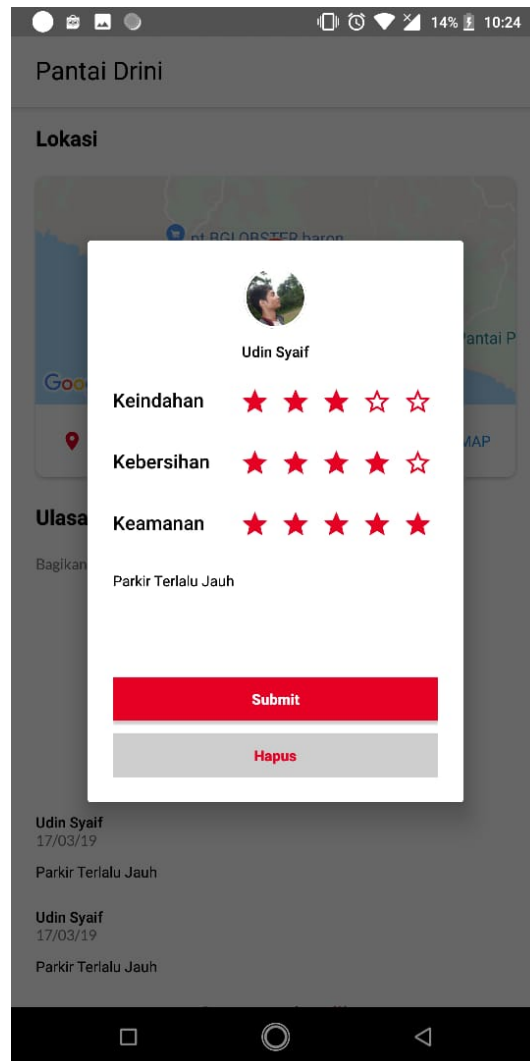
Halaman ini menampilkan gambar galeri sebuah objek wisata secara *fullscreen*. Halaman antarmuka ini terlihat seperti pada Gambar 5.21.



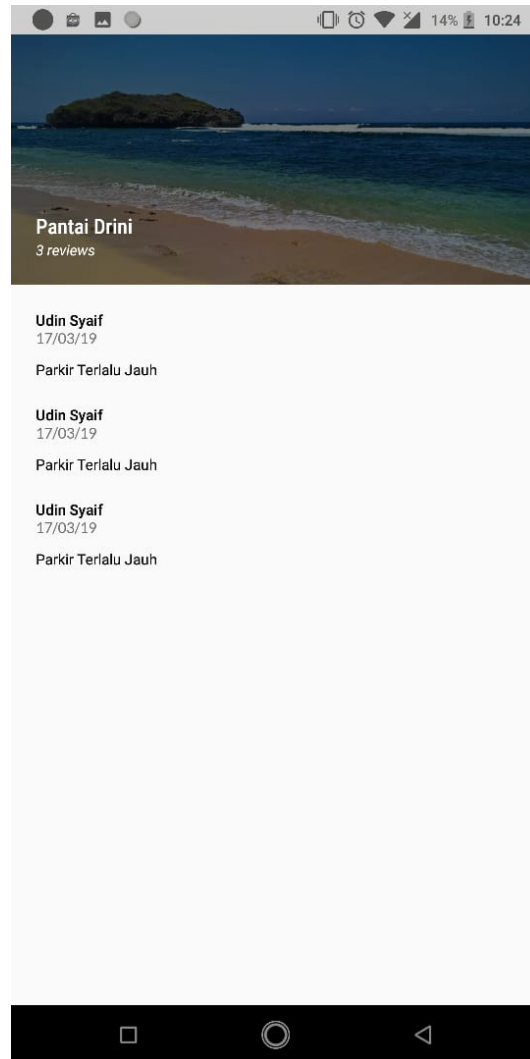
Gambar 5.17 Halaman Detail Objek Wisata



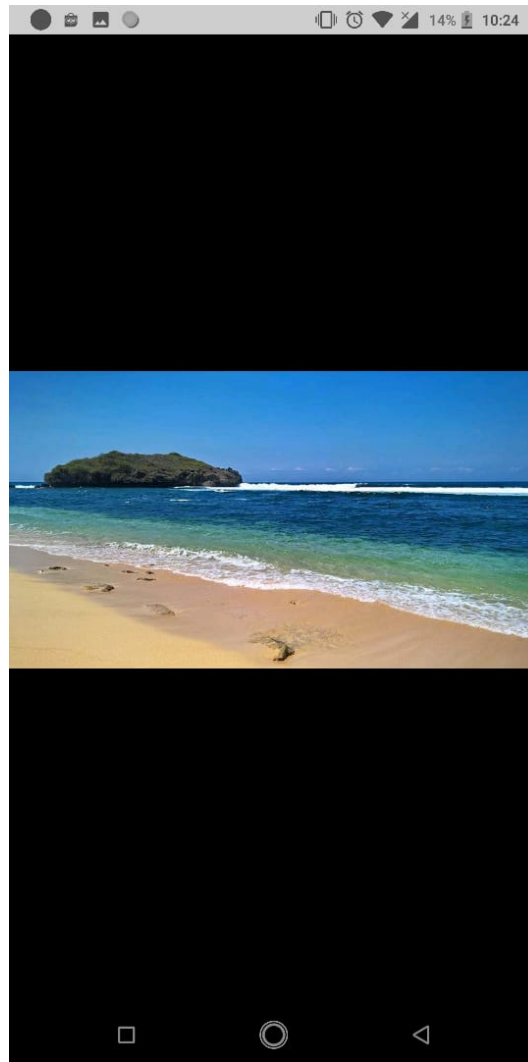
Gambar 5.18 Halaman Detail Objek Wisata((Lanjutan))



Gambar 5.19 Halaman *Edit* atau Tambah Penilaian Objek Wisata



Gambar 5.20 Halaman Lihat Seluruh Komentar Objek Wisata



Gambar 5.21 Halaman Lihat Gambar Galeri Objek Wisata

5.2. Percobaan Kasus

Percobaan kasus dilakukan untuk mengetahui apakah penilaian menggunakan metode *TOPSIS* yang dilakukan sistem sesuai dengan perhitungan manual dengan metode yang sama. Berikut ini langkah perhitungan metode *TOPSIS* yang dilakukan secara manual.

1. Terdapat 10 data alternatif objek wisata, misalnya objek wisata Air Terjun Banyunibo dengan nilai kriteria jarak(C1) 1.0, kriteria HTM(C2) 1.0, kriteria fasilitas(C3) 1.0, kriteria keamanan(C4) 2.0, kriteria kebersihan(C5) 5.0, kriteria keindahan(C6)

5.0 dan kriteria transportasi(c7) 3.0. Sedangkan data alternatif selengkapnya pada percobaan ini disajikan dalam Tabel 5.1.

Tabel 5.1 Tabel Matriks Nilai Alternatif

Objek Wisata	C1	C2	C3	C4	C5	C6	C7
Air Terjun Banyunibo	1.0	1.0	1.0	2.0	5.0	5.0	3.0
Air Terjun Kedung Kandang	2.0	1.0	2.0	4.0	4.0	4.0	3.0
Air Terjun Lepo	2.0	1.0	3.0	2.0	3.0	5.0	3.0
Air Terjun Setawing	2.0	1.0	3.0	3.0	4.0	5.0	4.0
Alun-alun Kidul	1.0	1.0	2.0	5.0	3.0	4.0	5.0
Bendung Tegal	1.0	1.0	3.0	3.0	2.0	4.0	5.0
Bukit Bintang Patuk	1.0	1.0	5.0	3.0	3.0	5.0	5.0
Bukit Panguk Kediwung	1.0	1.0	3.0	4.0	3.0	5.0	2.0
Bukit Paralayang Parangtritis	2.0	1.0	3.0	3.0	3.0	5.0	3.0
Candi Banyunibo	1.0	1.0	2.0	4.0	4.0	4.0	4.0

2. *User* memasukkan skala kepentingan kriteria seperti pada Tabel 5.2.

Tabel 5.2 Tabel Kriteria yang Diinputkan *User*

Kriteria	Keterangan	Nilai
Keindahan	<i>Benefit</i>	5.0
Keamanan	<i>Benefit</i>	3.0

Tabel 5.3 Tabel Kriteria yang Diinputkan *User* (Lanjutan)

Kebersihan	<i>Benefit</i>	4.0
Fasilitas	<i>Benefit</i>	4.0
Transportasi	<i>Benefit</i>	3.0
HTM	<i>Cost</i>	5.0
Jarak	<i>Cost</i>	5.0

3. Dilakukan normalisasi nilai pada matriks nilai alternatif. Sesuai Rumus 4.1, perhitungan normalisasi objek wisata Air Terjun Banyunibo pada kriteria jarak(C1) adalah dengan membagi nilai kriteria jarak objek wisata tersebut dengan akar jumlah kuadrat dari kriteria jarak pada 10 kriteria yang ada. Misalnya normalisasi kriteria jarak(C1) pada alternatif Air Terjun Banyunibo adalah nilai kriteria jarak(C1) dibagi dengan akar jumlah kuadrat dari 10 kriteria, yaitu $5/14.1 = 0.354$. Selanjutnya, dilakukan perhitungan yang sama pada setiap kriteria alternatif. Hasil normalisasi disajikan pada Tabel 5.4.
4. Dilakukan pembobotan pada matriks yang telah ternormalisasi sesuai dengan Rumus 4.2, yaitu perkalian antara matriks ternormalisasi dengan kriteria yang telah dimasukkan *user*. Misalnya normalisasi terbobot kriteria jarak(C1) pada alternatif Air Terjun Banyunibo adalah nilai ternormalisasi kriteria jarak(C1) dikali dengan nilai kriteria jarak(C1) yang dimasukkan *user*, yaitu $0.354 \times 5.0 = 1.772$. Selanjutnya, dilakukan perhitungan yang sama pada setiap kriteria alternatif. Hasilnya disajikan dalam Tabel 5.5.

Tabel 5.4 Tabel Normalisasi Matriks Kriteria

Objek Wisata	C1	C2	C3	C4	C5	C6	C7
Air Terjun Banyu- nibo	0.213	0.316	0.110	0.185	0.453	0.342	0.247
Air Terjun Kedung Kandang	0.426	0.316	0.220	0.370	0.362	0.273	0.247
Air Terjun Lepo	0.426	0.316	0.329	0.185	0.272	0.342	0.247
Air Terjun Seta- wing	0.426	0.316	0.329	0.277	0.362	0.342	0.330
Alun-alun Kidul	0.213	0.316	0.220	0.462	0.272	0.273	0.412
Bendung Tegal	0.213	0.316	0.329	0.277	0.181	0.273	0.412
Bukit Bintang Pa- tuk	0.213	0.316	0.549	0.277	0.272	0.342	0.412
Bukit Panguk Kedi- wung	0.213	0.316	0.329	0.370	0.272	0.342	0.165
Bukit Paralayang Parangtritis	0.426	0.316	0.329	0.277	0.272	0.342	0.247
Candi Banyunibo	0.213	0.316	0.220	0.370	0.362	0.273	0.330

Tabel 5.5 Tabel Matriks Y

Objek Wisata	C1	C2	C3	C4	C5	C6	C7
Air Terjun Banyu- nibo	1.066	1.581	0.439	0.555	1.811	1.709	0.742

Tabel 5.6 Tabel Matriks Y(Lanjutan)

Objek Wisata	C1	C2	C3	C4	C5	C6	C7
Air Terjun Kedung Kandang	2.132	1.581	0.878	1.109	1.449	1.367	0.742
Air Terjun Lepo	2.132	1.581	1.317	0.555	1.086	1.709	0.742
Air Terjun Seta-wing	2.132	1.581	1.317	0.832	1.449	1.709	0.990
Alun-alun Kidul	1.066	1.581	0.878	1.387	1.086	1.367	1.237
Bendung Tegal	1.066	1.581	1.317	0.832	0.724	1.367	1.237
Bukit Bintang Patuk	1.066	1.581	2.195	0.832	1.086	1.709	1.237
Bukit Panguk Keduwung	1.066	1.581	1.317	1.109	1.086	1.709	0.495
Bukit Paralayang Parangtritis	2.132	1.581	1.317	0.832	1.086	1.709	0.742
Candi Banyunibo	1.066	1.581	0.878	1.109	1.449	1.367	0.990

5. Setelah didapat matriks ternormalisasi terbobot, selanjutnya adalah mencari nilai solusi ideal positif dan negatif. Nilai solusi ideal positif adalah nilai *max* kriteria dari seluruh alternatif jika kriteria bersifat *benefit* dan *min* kriteria jika kriteria bersifat *cost*. Sementara nilai solusi ideal negatif adalah nilai *max* kriteria dari seluruh alternatif jika kriteria bersifat *cost* dan *min* kriteria jika kriteria bersifat *benefit*. Hasilnya disajikan dalam Tabel 5.7.

Tabel 5.7 Tabel Nilai Solusi Ideal

	C1	C2	C3	C4	C5	C6	C7
A^+	1.066	1.581	2.195	1.387	1.811	1.709	1.237
A^-	2.132	1.581	0.439	0.555	0.724	1.367	0.495

6. Lalu dilakukan perhitungan jarak alternatif dengan solusi ideal positif dan negatif sesuai dengan Rumus 4.5 dan Rumus 4.6. Perhitungan jarak alternatif terhadap solusi ideal positif/negatif adalah akar jumlah kuadrat seluruh kriteria dari selisih antara nilai ternormalisasi terbobot dengan nilai solusi ideal positif/negatif. Hasil perhitungannya disajikan dalam Tabel 5.8.

Tabel 5.8 Tabel Matriks Jarak Alternatif Terhadap Solusi Ideal

Objek Wisata	Jarak Solusi Ideal Positif	Jarak Solusi Ideal Negatif
Air Terjun Banyu- nibo	2.005	1.579
Air Terjun Kedung Kandang	1.855	1.042
Air Terjun Lepo	1.836	1.039
Air Terjun Seta- wing	1.552	1.317
Alun-alun Kidul	1.542	1.644
Bendung Tegal	1.541	1.592
Bukit Bintang Pa- tuk	0.912	2.258

Tabel 5.9 Tabel Matriks Jarak Alternatif Terhadap Solusi Ideal (Lanjutan)

Objek Wisata	Jarak Solusi Ideal Positif	Jarak Solusi Ideal Negatif
Bukit Panguk Keduwung	1.387	1.569
Bukit Paralayang Parangtritis	1.728	1.076
Candi Banyunibo	1.456	1.551

7. Setelah didapat jarak alternatif dengan nilai solusi ideal positif dan negatif. Maka dapat dihitung kedekatan alternatif terhadap solusi ideal sesuai dengan Rumus 4.7. Yaitu nilai jarak solusi ideal negatif dibagi dengan jumlah jarak solusi ideal positif dan jarak solusi ideal negatif. Misalnya kedekatan alternatif terhadap solusi ideal objek wisata Air Terjun Banyunibo adalah $1.165/(2.456+1.165) = 0.322$. Selanjutnya dilakukan perhitungan yang sama terhadap seluruh alternatif. Hasil perhitungan kedekatan alternatif dengan solusi ideal disajikan dalam Tabel 5.10.

Tabel 5.10 Tabel Matriks Kedekatan Alternatif terhadap Solusi Ideal

Objek Wisata	Kedekatan Terhadap Solusi Ideal
Air Terjun Banyunibo	0.441
Air Terjun Kedung Kandang	0.360
Air Terjun Lepo	0.362
Air Terjun Setawing	0.459
Alun-alun Kidul	0.516

Tabel 5.11 Tabel Matriks Kedekatan Alternatif terhadap Solusi Ideal (Lanjutan)

Objek Wisata	Kedekatan Terhadap Solusi Ideal
Bendung Tegal	0.508
Bukit Bintang Patuk	0.712
Bukit Panguk Kediwung	0.531
Bukit Paralayang Parangtritis	0.384
Candi Banyunibo	0.516

Sementara dengan data yang sama, hasil perhitungan *TOPSIS* yang dilakukan oleh sistem terlihat seperti pada Gambar 5.22.

```
Air Terjun Banyunibo : 0.441
Air Terjun Kedung Kandang : 0.360
Air Terjun Lepo : 0.362
Air Terjun Setawing : 0.459
Alun-alun Kidul : 0.516
Bendung Tegal : 0.508
Bukit Bintang Patuk : 0.712
Bukit Panguk Kediwung : 0.531
Bukit Paralayang Parangtritis : 0.384
Candi Banyunibo : 0.516
```

Gambar 5.22 Hasil Perhitungan *TOPSIS* yang Dilakukan Sistem

Dapat dilihat, bahwa perhitungan yang dilakukan sistem dengan perhitungan yang dilakukan manual menghasilkan nilai yang sama. Dengan demikian, dapat disimpulkan bahwa sistem berhasil menerapkan metode *TOPSIS* dan layak digunakan. Selain itu, tentu saja perhitungan yang dilakukan sistem jauh lebih cepat daripada perhitungan manual.

5.3. Pengujian Sistem

Pengujian sistem merupakan tahapan akhir dari penelitian ini. Tujuan tahap pengujian ini adalah untuk memastikan elemen-elemen dari sistem berfungsi dengan baik

sesuai dengan yang diharapkan dan sesuai dengan perancangan sistem yang telah dibuat sebelumnya.

5.3.1. Pengujian *Alpha*

Tahapan pertama dalam pengujian dengan metode *black box* adalah pengujian *alpha*. Pengujian *alpha* terfokus pada persyaratan fungsionalitas sistem dengan cara uji coba data oleh pengembang sistem, yaitu dengan memasukkan data yang benar dan data yang salah. Adapun rencana pengujian *alpha* dapat dilihat pada Tabel 5.12.

Tabel 5.12 Rencana pengujian *alpha*

No	Pengujian	Detail Pengujian
1	<i>Login dan log out admin atau user</i>	Verifikasi <i>login</i> dan <i>log out</i> admin atau <i>user</i>
2	Mengelola data objek wisata oleh admin	Tambah, <i>edit</i> dan hapus
3	Ganti <i>password</i> admin	Mengganti <i>password</i> untuk admin
4	<i>User</i> mengelola ulasan	Tambah, <i>edit</i> dan hapus
5	Pemilihan kriteria objek wisata oleh <i>user</i>	Masukkan kriteria, ubah dan <i>reset</i>

Berdasarkan hasil pengujian *alpha* terhadap sistem, dapat disimpulkan bahwa sistem yang dihasilkan berhasil dan dapat diterima. Hal ini didasarkan karena sistem telah memenuhi kebutuhan awal dan menghasilkan data keluaran seperti yang diharapkan pengembang sistem.

5.3.2. Pengujian *Beta*

Pengujian *beta* adalah pengujian yang dilakukan peneliti secara obyektif dengan terjun langsung ke lapangan untuk mengetahui pendapat dari responden mengenai sistem yang telah dibuat.

1. Pengujian Fungsional Sistem

Pengujian fungsional adalah pengujian apakah fungsi-fungsi yang ada pada sistem berjalan dengan baik dan benar. Pengujian fungsional pada sistem ini dapat dilihat pada Tabel 5.13.

Tabel 5.13 Rencana pengujian fungsional

No	Pernyataan	Ya	Tidak
1	Proses <i>login</i> dan <i>log out</i> berjalan dengan baik dan benar		
2	Sistem dapat mengelola (tambah, <i>edit</i> dan hapus) objek wisata dengan baik		
3	Sistem dapat mengelola (tambah, <i>edit</i> dan hapus) ulasan objek wisata dengan baik		

Tabel 5.14 Rencana pengujian fungsional (Lanjutan)

No	Pernyataan	Ya	Tidak
4	Sistem dapat melakukan proses perhitungan metode <i>TOPSIS</i> dengan tepat		
5	Sistem dapat menampilkan rekomendasi objek wisata sesuai dengan kriteria yang ditentukan oleh <i>user</i>		

2. Pengujian Usabilitas Sistem

Pengujian fungsional adalah pengujian apakah fungsi-fungsi yang ada pada sistem berjalan dengan baik dan benar. Pengujian fungsional pada sistem ini dapat dilihat pada Tabel 5.15.

Tabel 5.15 Rencana pengujian usabilitas

No	Pernyataan	SS	S	N	TS	STS
1	Sistem memiliki tampilan antarmuka yang menarik dan mudah digunakan					
2	Sistem memiliki kemudahan saat mengelola data					
3	Sistem memiliki <i>error handling</i> saat mengelola data					
4	Sistem memberi kemudahan saat memasukkan kriteria objek wisata					

Tabel 5.16 Rencana pengujian usabilitas (Lanjutan)

No	Pernyataan	SS	S	N	TS	STS
5	Sistem memberikan rekomendasi objek wisata sesuai kriteria yang dimasukkan					
6	Sistem memberikan rekomendari objek wisata sesuai kriteria yang dimasukan					

Keterangan :

SS : Sangat Setuju

S : Setuju

N : Netral

TS : Tidak Setuju

STS : Sangat Tidak Setuju

BAB VI

HASIL PENELITIAN DAN PEMBAHASAN

6.1. Proses Pengembangan dengan Metode *Extreme Programming*

Pengembangan sistem ini menggunakan metode *Extreme Programming*. Seperti yang dijelaskan pada bab sebelumnya, metode ini memiliki 4 tahapan, yaitu: *planning*, *design*, *coding*, *testing*.

1. Tahap *Planning* 1

Pada tahap perencanaan pertama, penulis melakukan observasi untuk memperoleh kebutuhan sistem. Penulis melakukan observasi terhadap sistem yang sudah berjalan maupun terhadap penelitian-penelitian sebelumnya yang berkaitan dengan penelitian kali ini. Hasil dari observasi pertama ini, sistem yang dibuat harus dapat mengelola data objek wisata yang ada di Daerah Istimewa Yogyakarta, dan sistem harus dapat menampilkan hasil rekomendasi objek wisata dengan tampilan yang sederhana.

2. Tahap *Design* 1

Setelah mendapat kebutuhan sistem, lalu penulis mulai merancang desain *database*, desain diagram alur sistem (UML) dan desain antarmuka. Dari tahap ini, dihasilkan bahwa *database* akan dibuat pada *MySQL* dan harus menggunakan *API* untuk menghubungkan aplikasi.

3. Tahap *Coding* 1

Setelah tahap perancangan selesai, penulis melakukan implementasi yaitu *coding*. 2 aplikasi yang dilakukan implementasi yaitu adalah *CMS (Content Management*

System untuk mengelola data dan aplikasi *Explore Jogja* untuk menampilkan data objek wisata yang ada kepada *user*(wisatawan). Penulis juga melakukan implementasi metode *TOPSIS* yang dihitung secara otomatis setelah *user* memasukkan kriteria.

4. Tahap *Testing 1*

Setelah berhasil melakukan tahap *coding* selanjutnya penulis melakukan *testing*. Dari tahap ini, penulis mendapat respon untuk menambah fitur pencarian objek wisata pada aplikasi *mobile* serta menambahkan validasi *input* pada *CMS (Content Management System)*.

5. Tahap *Planning 2*

Setelah mendapatkan masukan, penulis melakukan perencanaan kedua. Pada perencanaan ini penulis merencanakan mengubah sedikit tampilan antarmuka untuk memasukkan fitur pencarian objek wisata. Pada bagian *CMS* penulis menambahkan beberapa baris *code* untuk melakukan validasi *input* agar tidak ada data yang kosong.

6. Tahap *Design 2*

Yang diubah desain pada tahap ini adalah tampilan antarmuka pada aplikasi *mobile* agar kolom pencarian objek wisata dapat digunakan dan menambahkan teks validasi agar saat pengisian data pada *CMS* tidak bisa kosong.

7. Tahap *Coding 2*

Pada tahap ini, penulis melakukan implementasi sesuai dengan apa yang sudah direncanakan dan didesain pada tahap sebelumnya untuk menambah fitur pencarian objek wisata dan validasi pada *input*.

8. Tahap *Testing* 2

Setelah berhasil melakukan tahap *coding* selanjutnya penulis melakukan *testing*.

Dari tahap ini, penulis mendapat masukan bagaimana jika dari salah satu kriteria tidak ingin digunakan oleh *user*. Maka seharusnya disediakan pilihan.

9. Tahap *Planning* 3

Setelah mendapat masukan, penulis ingin menambahkan pilihan jika *user* tidak ingin menggunakan salah satu kriteria.

10. Tahap *Design* 3

Penulis mengubah kembali tampilan antarmuka pada aplikasimobile agar *user* dapat tidak menggunakan salah satu kriteria yang ada.

11. Tahap *Coding* 3

Penulis mengubah beberapa logika dan tampilan antarmuka dalam perhitungan *TOPSIS* jika *user* ingin tidak menggunakan salah satu kriteria yang ada.

12. Tahap *Testing* 3

Setelah implementasi, sudah tidak ada masukan yang diterima oleh penulis. Sehingga sistem sudah dapat dikatakan berhasil diimplementasikan.

6.2. Deskripsi Sistem

Sistem Pendukung Keputusan Pemilihan Obyek Wisata di Yogyakarta Menggunakan Metode Topsis Berbasis Android adalah sistem yang membantu para wisatawan yang ingin mengunjungi objek-objek wisata di Daerah Istimewa Yogyakarta. Sistem ini terdiri dari dua aplikasi, yaitu yang digunakan admin untuk memasukkan data-data objek

wisata, dan sistem yang digunakan *user* untuk mencari rekomendasi objek wisata sesuai kriteria yang mereka inginkan.

Metode yang digunakan untuk mencari rekomendasi objek wisata adalah metode *TOPSIS*. Metode ini menghitung alternatif terbaik dengan cara mencari alternatif yang nilainya paling dekat dengan nilai solusi ideal positif dan paling jauh dengan nilai solusi ideal negatif. Proses perhitungan ini dilakukan saat *user* memasukkan kriteria yang mereka inginkan. Setelah proses perhitungan selesai, sistem akan mengurutkan objek wisata dengan nilai akhir *TOPSIS* dari yang terbesar ke yang terkecil. Setelah itu *user* bisa menentukan objek wisata mana yang akan dikunjungi.

6.3. Data Masukan (*input*) Sistem

Data masukan yang diperlukan untuk melakukan perhitungan *TOPSIS* pada sistem ini adalah kriteria yang dimasukkan oleh *user* dan data nilai kriteria alternatif yang dimiliki oleh objek wisata. Kriteria yang diperhitungkan disini adalah kemudahan transportasi, kelengkapan fasilitas, kebersihan, keindahan, keamanan, harga tiket masuk dan jarak wisatawan dari objek wisata. Ada pula tujuh kriteria tadi didapat dari objek wisata maupun dari *user*. Kriteria yang didapat dari objek wisata adalah kemudahan transportasi, kelengkapan fasilitas dan harga tiket masuk. Kriteria jarak didapat dari lokasi *user*, sementara kriteria kebersihan, keamanan, dan keindahan didapat dari rata-rata penilaian *user* terhadap objek wisata.

6.4. Data Keluaran (*output*) Sistem

Data keluaran yang dihasilkan sistem ini berupa urutan objek wisata berdasarkan nilai dari metode *TOPSIS* terhadap alternatif sesuai dengan kriteria yang dimasukkan oleh *user*. Objek wisata yang direkomendasikan oleh sistem adalah objek wisata yang

memiliki nilai paling besar.

6.5. Pembahasan Pengujian Sistem

6.5.1. Hasil Pengujian Fungsionalitas dan Usabilitas Sistem

Pengujian ini melibatkan 30 orang responden. Adapun daftar penguji sistem ini disajikan pada Tabel 6.1.

Tabel 6.1 Daftar Responden

No	Kepentingan	Pekerjaan	Bidang	Jumlah
1	Admin	Software Engineer	IT	7
2	Admin dan <i>User</i>	Mahasiswa	IT	8
3	<i>User</i>	Wisatawan	Umum	15
Jumlah				30

6.5.2. Hasil dan Pembahasan Pengujian *Alpha*

Berdasarkan pengujian *alpha*, secara keseluruhan sistem yang dibangun dapat dibidang berhasil atau diterima. Karena secara fungsionalitas, sistem dapat dijalankan sesuai dengan apa yang diharapkan peneliti.

6.5.3. Hasil dan Pembahasan Pengujian *Beta*

Pengujian dilakukan terhadap fungsionalitas dan usabilitas sistem. Hasil pengujian fungsionalitas dapat dilihat pada Tabel 6.2 sementara pengujian usabilitas dapat dilihat pada Tabel 6.4.

Persentase pengujian fungsionalitas sistem adalah :

a. Menjawab YA : $(150/150) \times 100\% = 100\%$

b. Menjawab TIDAK : $(0/150) \times 100\% = 0\%$

Persentase pengujian usabilitas sistem adalah :

a. Menjawab Sangat Setuju : $(67/180) \times 100\% = 37.2\%$

b. Menjawab Setuju : $(106/180) \times 100\% = 58.8\%$

c. Menjawab Netral : $(6/180) \times 100\% = 3.3\%$

d. Menjawab Tidak Setuju : $(1/180) \times 100\% = 0.5\%$

e. Menjawab Sangat Tidak Setuju : $(0/180) \times 100\% = 0\%$

Tabel 6.2 Hasil Pengujian Fungsionalitas Sistem

No	Pernyataan	Ya	Tidak
1	Proses <i>login</i> dan <i>log out</i> berjalan dengan baik dan benar	30	0
2	Sistem dapat mengelola (tambah, <i>edit</i> dan hapus) objek wisata dengan baik	30	0
3	Sistem dapat mengelola (tambah, <i>edit</i> dan hapus) ulasan objek wisata dengan baik	30	0
4	Sistem dapat melakukan proses perhitungan metode <i>TOPSIS</i> dengan tepat	30	0

Tabel 6.3 Hasil Pengujian Fungsionalitas Sistem (Lanjutan)

No	Pernyataan	Ya	Tidak
5	Sistem dapat menampilkan rekomendasi objek wisata sesuai dengan kriteria yang ditentukan oleh <i>user</i>	30	0

Hasil dari pengujian fungsionalitas sistem menunjukkan bahwa 100% responden menyatakan fungsionalitas sistem telah berjalan dengan baik, yang artinya sistem sudah berjalan sesuai dengan apa yang diharapkan peneliti.

Tabel 6.4 Hasil Pengujian Usabilitas Sistem

No	Pernyataan	SS	S	N	TS	STS
1	Sistem memiliki tampilan antarmuka yang menarik dan mudah digunakan	18	12	0	0	0
2	Sistem memiliki kemudahan saat mengelola data	5	24	1	0	0
3	Sistem memiliki <i>error handling</i> saat mengelola data	6	21	2	1	0
4	Sistem memberi kemudahan saat memasukkan kriteria objek wisata	10	19	1	0	0
5	Sistem memberikan rekomendasi objek wisata sesuai kriteria yang dimasukkan	11	18	1	0	0
6	Sistem memberikan rekomendasi objek wisata sesuai kriteria yang dimasukkan	17	12	1	0	0

Berdasarkan pengujian usabilitas sistem, diperoleh kesimpulan bahwa sebagian besar responden setuju dengan sistem yang telah dibangun. Data hasil pengujian usabilitas menyatakan bahwa 37.2% responden sangat setuju, 58.8% responden menyatakan setuju, 3.3%

responden menyatakan netral, 0.5% responden menyatakan tidak setuju, dan 0.0% responden menyatakan sangat tidak setuju.

6.6. Percobaan Perhitungan Metode *TOPSIS* pada Sistem

Setelah sistem berhasil dibangun, maka dilakukan 5 kali percobaan perhitungan sistem dari beberapa titik lokasi di Daerah Istimewa Yogyakarta dengan nilai kriteria yang dimasukkan *user* pun berbeda-beda. Percobaan 1 dilakukan untuk menguji perhitungan dengan nilai kriteria paling rendah. Percobaan 2 dimaksudkan untuk menguji perhitungan dengan nilai kriteria sembarang yang dimasukkan *user* tanpa perkiraan penentuan hasil. Percobaan 3 dimaksudkan untuk menguji perhitungan dengan nilai kriteria adalah nilai tengah yang disediakan oleh sistem. Percobaan 4 dimaksudkan untuk menguji perhitungan apabila salah satu nilai kriteria diunggulkan dari kriteria lainnya. Percobaan 5 dimaksudkan untuk menguji perhitungan apabila semua kriteria yang dimasukkan memiliki nilai yang tinggi.

1. Percobaan 1

Percobaan ini dilakukan di lokasi UIN Sunan Kalijaga Yogyakarta dengan kriteria yang dimasukkan adalah nilai terendah. Percobaan ini dimaksudkan untuk menguji perhitungan dengan nilai kriteria paling rendah. Kriteria yang digunakan pada percobaan ini disajikan pada Tabel 6.5.

Tabel 6.5 Tabel Kriteria yang Diinputkan *User* pada Percobaan 1

Kriteria	Nilai
Keindahan	1.0
Keamanan	1.0
Kebersihan	1.0

Tabel 6.6 Tabel Kriteria yang Diinputkan *User* pada Percobaan 1 (Lanjutan)

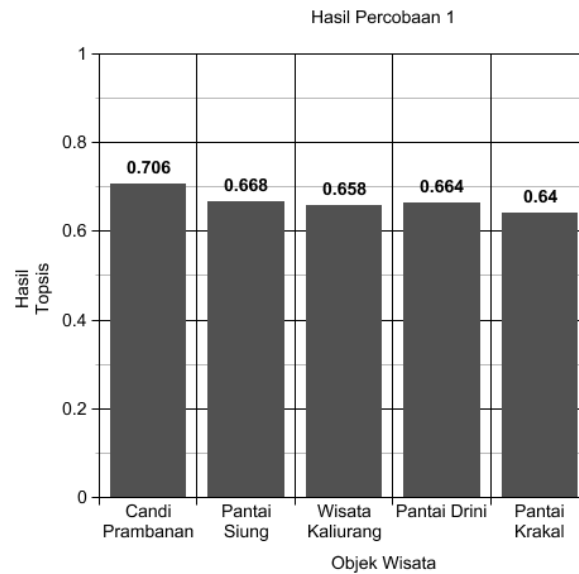
Kriteria	Nilai
Fasilitas	1.0
Transportasi	1.0
HTM	1.0
Jarak	1.0

Pada percobaan 1, rekomendasi wisata yang dihasilkan adalah objek wisata Candi Prambanan dengan nilai kriteria jarak(C1) 5.0, kriteria HTM(C2) 3.0, kriteria fasilitas(C3) 4.0, kriteria keamanan(C4) 5.0, kriteria kebersihan(C5) 4.0, kriteria keindahan(C6) 5.0 dan kriteria transportasi(c7) 5.0. Hasil 5 teratas rekomendasi objek wisata yang didapat disajikan pada tabel 6.7.

Tabel 6.7 5 Rekomendasi Objek Wisata pada Percobaan 1

Objek Wisata	C1	C2	C3	C4	C5	C6	C7	Hasil
Candi Prambanan	5.0	3.0	4.0	5.0	4.0	5.0	5.0	0.706
Pantai Siung	2.0	5.0	5.0	4.0	4.0	4.0	3.0	0.668
Wisata Kaliurang	4.0	5.0	5.0	4.0	4.0	4.0	4.0	0.658
Pantai Drini	3.0	5.0	5.0	3.5	4.0	3.2	4.0	0.644
Pantai Krakal	3.0	5.0	5.0	4.0	3.0	4.0	4.0	0.640

Grafik percobaan ini disajikan pada Gambar 6.1.



Gambar 6.1 Grafik Hasil Percobaan 1

2. Percobaan 2

Percobaan ini dilakukan di lokasi Masjid Agung Manunggal Bantul dengan kriteria yang dimasukkan adalah nilai acak. Percobaan ini dimaksudkan untuk menguji perhitungan dengan nilai kriteria sembarang yang dimasukkan *user* tanpa perkiraan penentuan hasil. Kriteria yang digunakan pada percobaan ini disajikan pada Tabel 6.8.

Tabel 6.8 Tabel Kriteria yang Diinputkan *User* pada Percobaan 2

Kriteria	Nilai
Keindahan	3.0
Keamanan	2.0
Kebersihan	4.0
Fasilitas	4.0
Transportasi	5.0

Tabel 6.9 Tabel Kriteria yang Diinputkan *User* pada Percobaan 2 (Lanjutan)

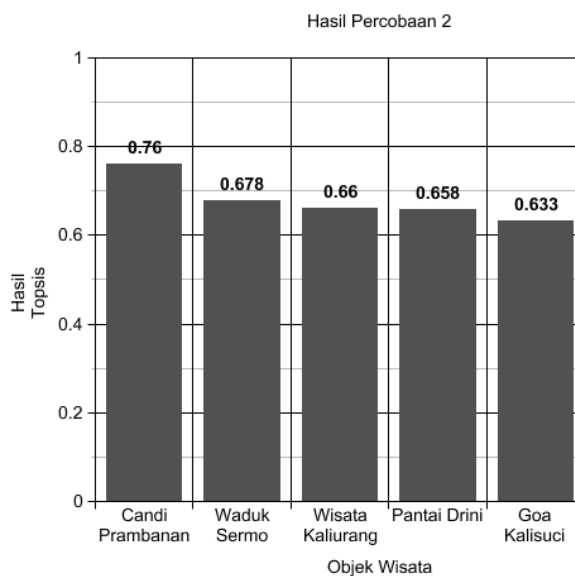
Kriteria	Nilai
HTM	4.0
Jarak	3.0

Pada percobaan 2, rekomendasi wisata yang dihasilkan adalah objek wisata Candi Prambanan dengan nilai kriteria jarak(C1) 5.0, kriteria HTM(C2) 3.0, kriteria fasilitas(C3) 4.0, kriteria keamanan(C4) 5.0, kriteria kebersihan(C5) 4.0, kriteria keindahan(C6) 5.0 dan kriteria transportasi(c7) 5.0. Hasil 5 teratas rekomendasi objek wisata yang didapat disajikan pada tabel 6.10.

Tabel 6.10 5 Rekomendasi Objek Wisata pada Percobaan 2

Objek Wisata	C1	C2	C3	C4	C5	C6	C7	Hasil
Candi Prambanan	4.0	3.0	4.0	5.0	4.0	5.0	5.0	0.760
Waduk Sermo	4.0	5.0	5.0	3.0	5.0	4.0	4.0	0.678
Wisata Kaliurang	4.0	5.0	5.0	4.0	4.0	4.0	4.0	0.660
Pantai Drini	3.0	5.0	5.0	3.5	4.0	3.2	4.0	0.658
Goa Kalisuci	3.0	2.0	4.0	3.0	3.0	4.0	4.0	0.633

Grafik percobaan ini disajikan pada Gambar 6.2.



Gambar 6.2 Grafik Hasil Percobaan 2

3. Percobaan 3

Percobaan ini dilakukan di lokasi Taman Deggung dengan kriteria yang dimasukkan adalah nilai tengah. Percobaan ini dimaksudkan untuk menguji perhitungan dengan nilai kriteria adalah nilai tengah yang disediakan oleh sistem. Kriteria yang digunakan pada percobaan ini disajikan pada Tabel 6.11.

Tabel 6.11 Tabel Kriteria yang Diinputkan *User* pada Percobaan 3

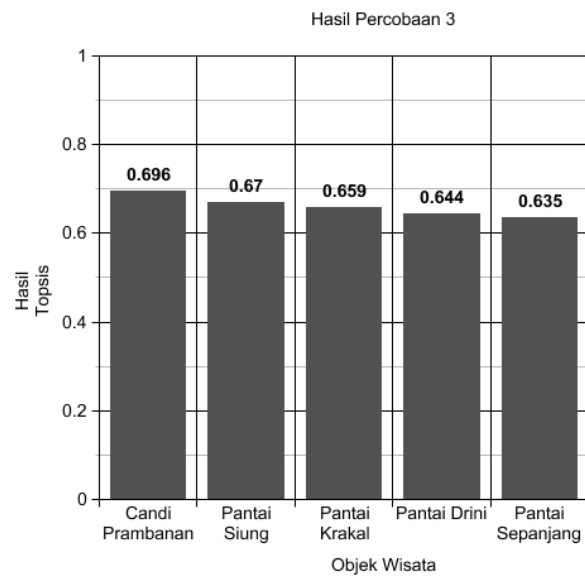
Kriteria	Nilai
Keindahan	3.0
Keamanan	3.0
Kebersihan	3.0
Fasilitas	3.0
Transportasi	3.0
HTM	3.0
Jarak	3.0

Pada percobaan 3, rekomendasi wisata yang dihasilkan adalah objek wisata Candi Prambanan dengan nilai kriteria jarak(C1) 5.0, kriteria HTM(C2) 3.0, kriteria fasilitas(C3) 4.0, kriteria keamanan(C4) 5.0, kriteria kebersihan(C5) 4.0, kriteria keindahan(C6) 5.0 dan kriteria transportasi(c7) 5.0. Hasil 5 teratas rekomendasi objek wisata yang didapat disajikan pada tabel 6.12.

Tabel 6.12 5 Rekomendasi Objek Wisata pada Percobaan 3

Objek Wisata	C1	C2	C3	C4	C5	C6	C7	Hasil
Candi Prambanan	5.0	3.0	4.0	5.0	4.0	5.0	5.0	0.696
Pantai Siung	2.0	5.0	5.0	4.0	4.0	4.0	3.0	0.670
Pantai Krakal	2.0	5.0	5.0	4.0	3.0	4.0	4.0	0.659
Pantai Drini	3.0	5.0	5.0	3.5	4.0	3.2	4.0	0.644
Pantai Sepanjang	3.0	5.0	4.0	4.0	4.0	5.0	3.0	0.635

Grafik percobaan ini disajikan pada Gambar 6.3.



Gambar 6.3 Grafik Hasil Percobaan 3

4. Percobaan 4

Percobaan ini dilakukan di lokasi Pusat Kuliner Belut Godean dengan nilai kriteria keindahan paling unggul dari kriteria lain. Percobaan ini dimaksudkan untuk menguji perhitungan apabila salah satu nilai kriteria diunggulkan dari kriteria lainnya. Kriteria yang digunakan pada percobaan ini disajikan pada Tabel 6.13.

Tabel 6.13 Tabel Kriteria yang Diinputkan *User* pada Percobaan 4

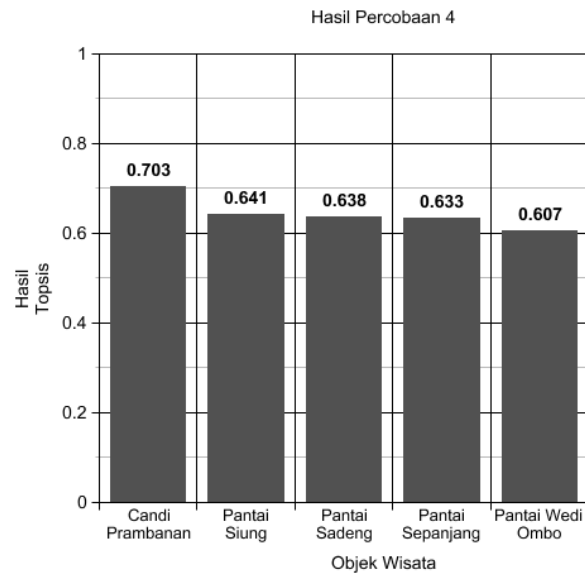
Kriteria	Nilai
Keindahan	5.0
Keamanan	3.0
Kebersihan	3.0
Fasilitas	1.0
Transportasi	3.0
HTM	2.0
Jarak	3.0

Pada percobaan 4, rekomendasi wisata yang dihasilkan adalah objek wisata Candi Prambanan dengan nilai kriteria jarak(C1) 5.0, kriteria HTM(C2) 3.0, kriteria fasilitas(C3) 4.0, kriteria keamanan(C4) 5.0, kriteria kebersihan(C5) 4.0, kriteria keindahan(C6) 5.0 dan kriteria transportasi(c7) 5.0. Hasil 5 teratas rekomendasi objek wisata yang didapat disajikan pada tabel 6.14.

Tabel 6.14 5 Rekomendasi Objek Wisata pada Percobaan 4

Objek Wisata	C1	C2	C3	C4	C5	C6	C7	Hasil
Candi Prambanan	4.0	3.0	4.0	5.0	4.0	5.0	5.0	0.703
Pantai Siung	2.0	5.0	5.0	4.0	4.0	4.0	3.0	0.641
Pantai Sadeng	1.0	5.0	5.0	3.0	4.0	5.0	3.0	0.638
Pantai Sepanjang	3.0	5.0	4.0	4.0	4.0	5.0	3.0	0.633
Pantai Wedi Ombo	2.0	5.0	4.0	3.0	4.0	5.0	3.0	0.607

Grafik percobaan ini disajikan pada Gambar 6.4.



Gambar 6.4 Grafik Hasil Percobaan 4

5. Percobaan 5

Percobaan ini dilakukan di lokasi Bandara Internasional Adi Sucipto Godean dengan rata-rata nilai kriteria tinggi. Percobaan ini dimaksudkan untuk menguji perhitungan apabila semua kriteria yang dimasukkan memiliki nilai yang tinggi. Kriteria yang digunakan pada percobaan ini disajikan pada Tabel 6.15.

Tabel 6.15 Tabel Kriteria yang Diinputkan *User* pada Percobaan 5

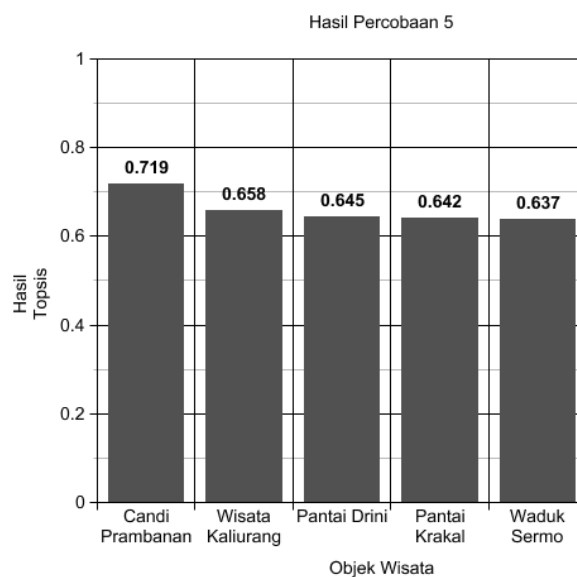
Kriteria	Nilai
Keindahan	4.0
Keamanan	4.0
Kebersihan	4.0
Fasilitas	5.0
Transportasi	5.0
HTM	5.0
Jarak	4.0

Pada percobaan 5, rekomendasi wisata yang dihasilkan adalah objek wisata Candi Prambanan dengan nilai kriteria jarak(C1) 5.0, kriteria HTM(C2) 3.0, kriteria fasilitas(C3) 4.0, kriteria keamanan(C4) 5.0, kriteria kebersihan(C5) 4.0, kriteria keindahan(C6) 5.0 dan kriteria transportasi(c7) 5.0. Hasil 5 teratas rekomendasi objek wisata yang didapat disajikan pada tabel 6.16.

Tabel 6.16 5 Rekomendasi Objek Wisata pada Percobaan 5

Objek Wisata	C1	C2	C3	C4	C5	C6	C7	Hasil
Candi Prambanan	5.0	3.0	4.0	5.0	4.0	5.0	5.0	0.719
Wisata Kaliurang	4.0	5.0	5.0	4.0	4.0	4.0	4.0	0.658
Pantai Drini	3.0	5.0	5.0	3.5	4.0	3.2	4.0	0.645
Pantai Krakal	3.0	5.0	5.0	4.0	3.0	4.0	4.0	0.642
Waduk Sermo	4.0	5.0	5.0	3.0	5.0	4.0	4.0	0.637

Grafik percobaan ini disajikan pada Gambar 6.5.



Gambar 6.5 Grafik Hasil Percobaan 5

Setelah dilakukan 5 kali percobaan di lokasi yang berbeda dan dengan kriteria yang berbeda pula, dapat dibagi menjadi beberapa penjelasan :

1. Objek Candi Prambanan selalu menjadi objek wisata yang direkomendasikan sistem. Hal ini disebabkan lokasi objek ini relatif dekat dengan ke lima titik lokasi percobaan. Selain itu, kriteria selain jarak dari objek lokasi ini juga memiliki rata-rata yang tinggi.
2. Saat salah satu nilai kriteria diunggulkan dari kriteria lain, maka objek wisata yang direkomendasikan adalah objek wisata yang memiliki nilai kriteria tinggi pada kriteria yang diunggulkan. Namun, bukan berarti kriteria lain tidak dipertimbangkan oleh sistem.
3. Walaupun salah satu objek wisata memiliki keunggulan pada salah satu kriterianya, dan kriteria yang dimasukkan oleh *user* paling tinggi pada kriteria tersebut

tidak akan menjadikan objek wisata tadi rekomendasi. Hal ini disebabkan karena sistem mempertimbangkan seluruh kriteria pada objek wisata, sehingga semakin besar keseluruhan nilai kriterianya maka objek wisata akan lebih sering muncul sebagai rekomendasi.

Kesimpulan dari hasil perhitungan 5 sampel percobaan diatas bisa dikatakan bahwa perhitungan menggunakan metode *TOPSIS* menghasilkan rekomendasi yang memiliki nilai keseluruhan kriteria tinggi. Semakin tinggi nilai keseluruhan kriteria pada objek wisata, maka akan sering objek wisata tersebut direkomendasikan oleh sistem. Objek wisata yang unggul hanya dalam salah satu kriteria tidak menjadi rekomendasi jika kriteria lainnya rendah meskipun kriteria yang paling dicari *user* adalah kriteria yang aling unggul tersebut. Dalam hal ini, ada baiknya Dinas Pariwisata DIY mengoptimalkan seluruh objek wisata yang ada di Daerah Istimewa Yogyakarta.