

RANCANGAN SISTEM REKOMENDASI PARIWISATA DI BOGOR BERBASIS WEB

Marcell Budiman ¹⁾ Wasino ²⁾ Lely Hiryanto ³⁾

¹⁾²⁾Program Studi Sistem Informasi, Fakultas Teknologi Informasi, Universitas Tarumanagara,
Jln. Letjen S. Parman No. 1, Jakarta, 11440, Indonesia
Email : marcellbudiman20@gmail.com¹⁾, wasino@fti.untar.ac.id²⁾

³⁾Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara
Jl. Letjen S. Parman No. 1, Jakarta Barat 11440, Indonesia
Email : lelyh@fti.untar.ac.id³⁾

ABSTRAK

Pariwisata adalah salah satu industri yang berkontribusi pada kemajuan ekonomi Indonesia. Wisatawan yang berwisata seringkali menghadapi rintangan untuk menyusun perjalanan wisata atau itinerary yang efisien. Rancangan aplikasi bernama Wistinerary ini bertujuan untuk membuat sebuah aplikasi berbasis web yang dapat memberikan rencana perjalanan wisata. Aplikasi ini menggunakan metode pengembangan Software Development Life Cycle dengan model pendekatan Waterfall. Dalam rancangan aplikasi ini menggunakan basis data yang bersumber dari penelitian sebelumnya dan diperluas untuk memenuhi kebutuhan. Aplikasi dikembangkan menggunakan beberapa tools yaitu MySQL, Python, Django dan Tailwind. Berdasarkan hasil pengujian, aplikasi telah berhasil diimplementasikan dengan baik sesuai dengan rancangan yang telah direncanakan.

Key words

Pariwisata, Aplikasi, Rute Perjalanan, Wisatawan, Itinerary

1. Pendahuluan

Pariwisata merupakan industri yang memberikan kontribusi pada kemajuan ekonomi Indonesia [1]. Indonesia memiliki potensi tinggi di bidang pariwisata karena Indonesia mempunyai kondisi alam yang indah dengan budaya yang beragam. Dengan berkembangnya sektor pariwisata, jumlah wisatawan pun turut bertambah. Wisatawan seringkali menghadapi tantangan dalam menyusun rencana perjalanan wisata atau *itinerary* yang efisien, terlebih jika wisatawan harus memilih destinasi dalam waktu yang terbatas. Wisatawan dapat mencari destinasi populer di suatu daerah, namun seringkali kesulitan dalam menentukan urutan rute perjalanan. Tingkat popularitas sebuah destinasi memegang peranan penting untuk memikat kedatangan wisatawan [2].

Rancangan aplikasi berbasis web bernama Wistinerary ini merupakan pengembangan dari perancangan sebelumnya [3], [4], [5]. Wistinerary dibuat untuk memberikan informasi mengenai destinasi wisata yang dapat membantu wisatawan dalam berwisata yang terinspirasi dari perancangan sebelumnya [6]. Dalam rancangan aplikasi Wistinerary ada beberapa batasan yang ditetapkan. Tujuan dari batasan ini adalah memastikan rancangan terfokuskan dan dapat diselesaikan. Batasan ini juga bertujuan menghindari pembuatan fitur yang berada di luar cakupan utama. Adapun batasan sistem yang dimaksud adalah:

1. Sistem hanya fokus pada penyusunan destinasi wisata dan tidak menangani fungsionalitas di luar itu seperti pemesanan tiket.
2. Aplikasi Wistinerary adalah aplikasi yang berbasis web yang dapat diakses melalui web, tidak mencakup pengembangan aplikasi *mobile*.
3. Sistem dikembangkan dengan menggunakan database MySQL, Python, Django dan Tailwind.

Berdasarkan rumusan masalah dan batasan yang telah ditetapkan, adapun tujuan yang ingin dicapai adalah sebagai berikut:

1. Menghasilkan rekomendasi rencana perjalanan yang optimal dengan mempertimbangkan jarak dan popularitas.
2. Membuat sebuah aplikasi berbasis web yang dapat memberikan rekomendasi rute perjalanan wisata berdasarkan destinasi yang dipilih.
3. Memberikan solusi dalam permasalahan pencarian rute perjalanan dengan menghemat biaya selama perjalanan.

Rancangan aplikasi ini diharapkan dapat memberikan manfaat terhadap berbagai pihak sebagai berikut:

1. Memberikan kemudahan untuk wisatawan dalam menyusun rencana perjalanan yang efisien dan sesuai preferensi.
2. Meningkatkan kualitas pengalaman berwisata dengan cara menghemat biaya perjalanan karena rute yang sudah optimal.

- Memberikan solusi dengan aplikasi berbasis web yang mudah diakses untuk merencanakan rencana perjalanan.

2. Metodologi

2.1. Metode Perancangan Aplikasi

Metode yang digunakan dalam perancangan aplikasi ini adalah *Software Development Life Cycle* dengan model pendekatan *Waterfall* yang dapat dilihat pada **Gambar 1**. Model ini digunakan karena sifatnya yang sistematis, artinya adalah setiap tahapan harus dilakukan saling berurutan. Setiap tahapan dalam metode ini harus diselesaikan sebelum tahapan berikutnya dilakukan. Model ini punya kelebihan karena pengembangannya dapat memberikan kualitas yang baik dan pelaksanaannya dilakukan secara bertahap [7].

1. Analisis (*Requirement*)

Tahapan analisis adalah tahapan yang memulai dan menjadi dasar dalam sebuah rancangan. Tahapan ini melakukan analisis tentang kebutuhan dari pengguna dan fitur yang harus dibuat. Selain itu pada tahapan ini juga menentukan data destinasi wisata yang dibutuhkan.

2. Perancangan (*Design*)

Setelah melakukan analisis semua kebutuhan yang ada, tahapan berikutnya adalah mengubah kebutuhan menjadi sebuah rancangan yang lebih detail. Rancangan ini akan mencakup tampilan dan struktur data yang dibutuhkan.

3. Implementasi (*Implementation*)

Tahapan ini memproses rancangan menjadi sebuah sistem program yang dapat berjalan. Proses ini mencakup pengembangan program dan implementasi algoritma dan tampilan aplikasi.

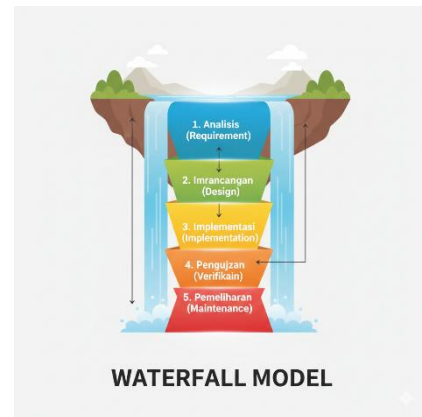
4. Pengujian (*Verifikasi*)

Tahapan pengujian merupakan tahapan penting setelah program selesai dibuat. Tahapan ini bertujuan untuk memastikan setiap fitur dapat berfungsi tanpa masalah dan aplikasi dapat memberikan rekomendasi rute perjalanan yang tepat.

5. Pemeliharaan (*Maintenance*)

Tahapan ini adalah tahapan terakhir dalam pembuatan aplikasi. Tahapan ini bertujuan untuk mendemonstrasikan dan mengevaluasi aplikasi. Pemeliharaan ini bertujuan untuk memperbaiki

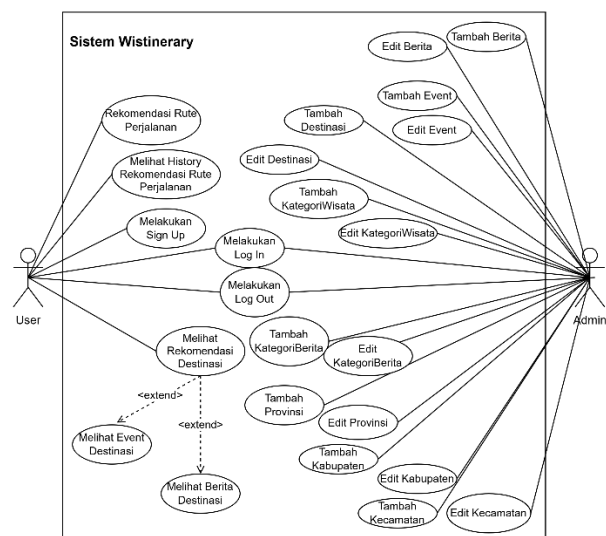
program ataupun penambahan fitur jika dibutuhkan.



Gambar 1 Analogi Metode Waterfall

2.2 Use Case Diagram

Dalam *Use Case Diagram*, individual yang berinteraksi dengan aplikasi disebut sebagai seorang aktor dan hubungan antara aktor dengan fungsi aplikasi akan digambarkan dengan garis. Tujuan dari diagram ini adalah membantu memahami bagaimana interaksi antara pengguna dengan sistem, sehingga aplikasi dapat dirancang dengan menyesuaikan kebutuhan pengguna [8]. Adapun *Use Case Diagram* pada perancangan aplikasi ini dapat dilihat pada **Gambar 2**. Pada aplikasi ini akan memiliki dua aktor yaitu *user* dan *admin*. Fungsi utama yang dapat dilakukan *user* adalah rekomendasi rute perjalanan dan melihat *history* rekomendasi rute perjalanan. Fitur lainnya adalah *user* dapat melihat rekomendasi berbagai macam informasi seperti destinasi wisata, kegiatan yang terkait dan juga berita terkait destinasi yang ada. Sedangkan dari sisi *admin*, fungsi utamanya adalah mengelola seluruh data yang ada. *admin* dapat melihat, mengubah, menghapus dan menambahkan data pada aplikasi.

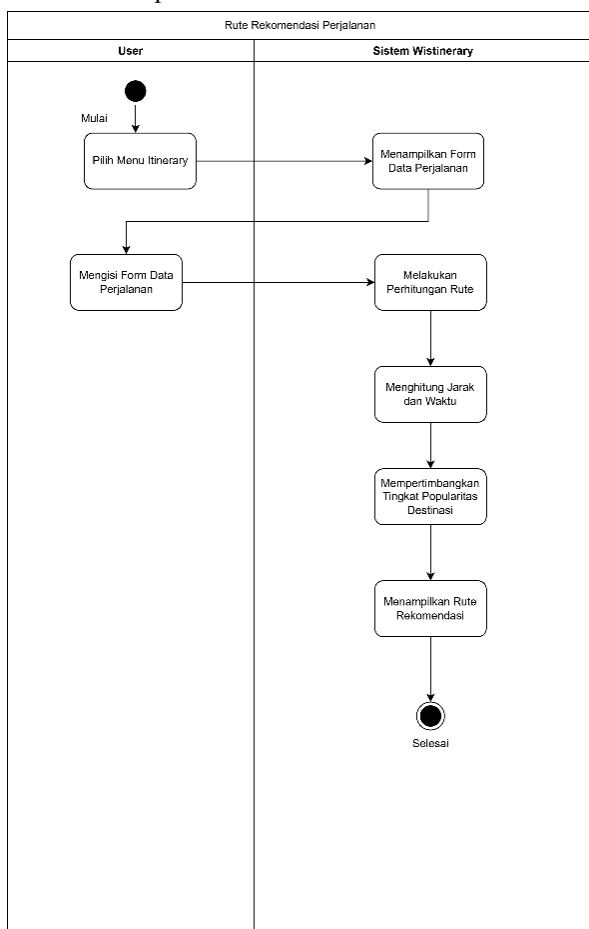


Gambar 2 Use Case Diagram

2.3 Activity Diagram

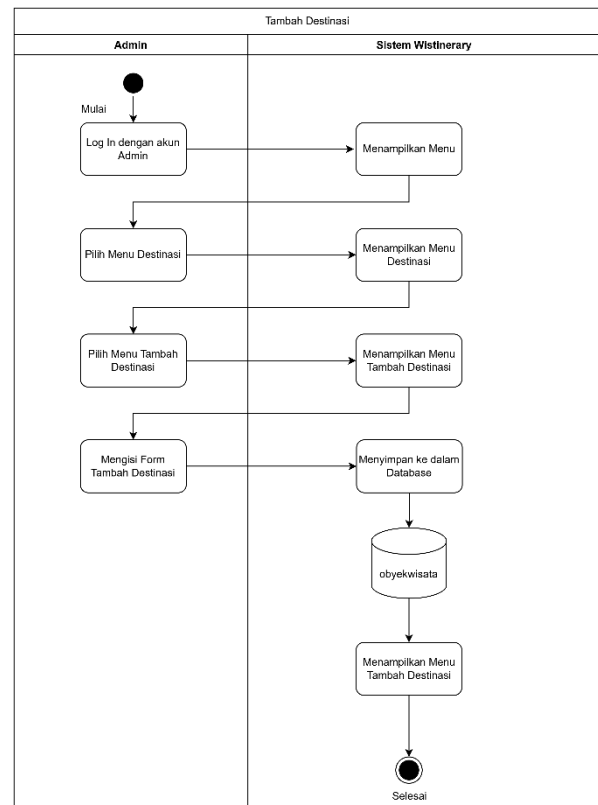
Activity diagram dapat memperlihatkan aliran proses antar aktivitas yang terjadi secara terstruktur. Dalam diagram ini, titik awal aktivitas digambarkan sebagai lingkaran hitam penuh sedangkan titik akhir aktivitas digambarkan sebagai lingkaran hitam penuh yang dikelilingi lingkaran lain. Proses yang terjadi digambarkan dengan persegi panjang yang dihubungkan dengan garis. Tujuan dari diagram ini adalah untuk memahami proses logika sistem secara detail sehingga memberikan pemahaman tentang fitur aplikasi dari awal hingga akhir [9].

Activity diagram fitur utama pada aplikasi ini yaitu rute rekomendasi perjalanan dapat dilihat pada **Gambar 3**. Proses ini dimulai ketika pengguna memilih menu *itinerary*. Ketika pengguna memilih menu *itinerary* maka pengguna akan melihat *form* data perjalanan. *Form* ini berfungsi sebagai tempat pengguna dapat memberikan informasi yang dibutuhkan untuk membuat rekomendasi rute perjalanan. Setelah pengguna mengisi *form* data perjalanan, sistem akan memproses yaitu dengan melakukan perhitungan rute berdasarkan input pengguna. Tahap terakhir adalah menampilkan hasilnya yaitu rekomendasi rute perjalanan dalam bentuk daftar destinasi serta peta interaktif.



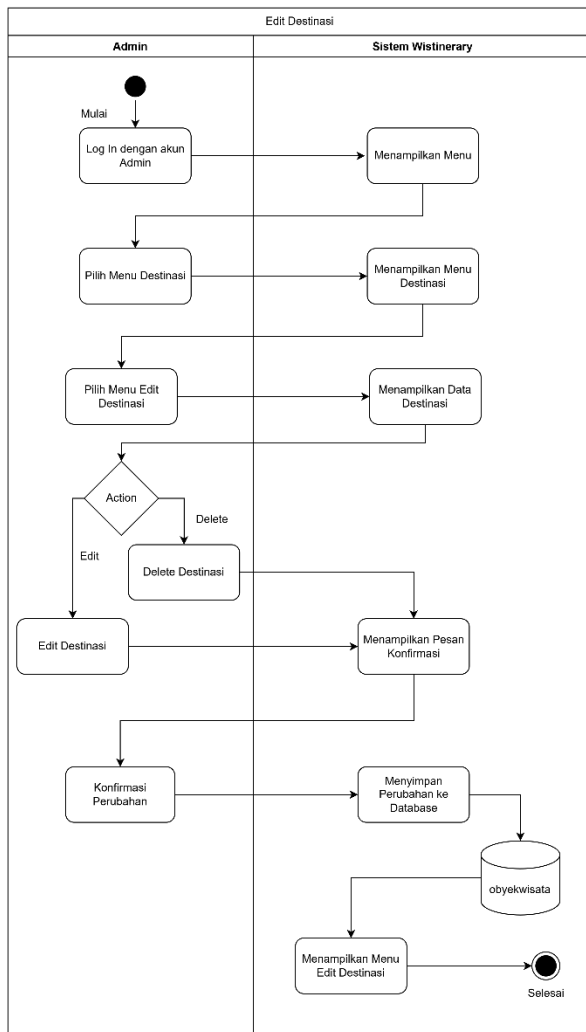
Gambar 3 Activity Diagram Rute Rekomendasi Perjalanan

Salah satu fitur lain adalah mengelola data yang dapat dilakukan oleh seorang *admin*. **Gambar 4** adalah *activity diagram* untuk proses penambahan data oleh *admin*. Proses ini dimulai ketika pengguna melakukan *login* dengan akun *admin*. *Admin* dapat memilih menu destinasi dan sistem akan menampilkan halaman destinasi dimana *admin* dapat menambahkan data baru. Untuk menambahkan data baru, *admin* harus mengisi *form* dengan data yang diperlukan. Setelah mengisi seluruh data, sistem akan menyimpan data ke dalam basis data. Kemudian sistem akan kembali menampilkan menu tambah destinasi dengan pesan bahwa data berhasil ditambahkan.



Gambar 4 Activity Diagram Tambah Destinasi

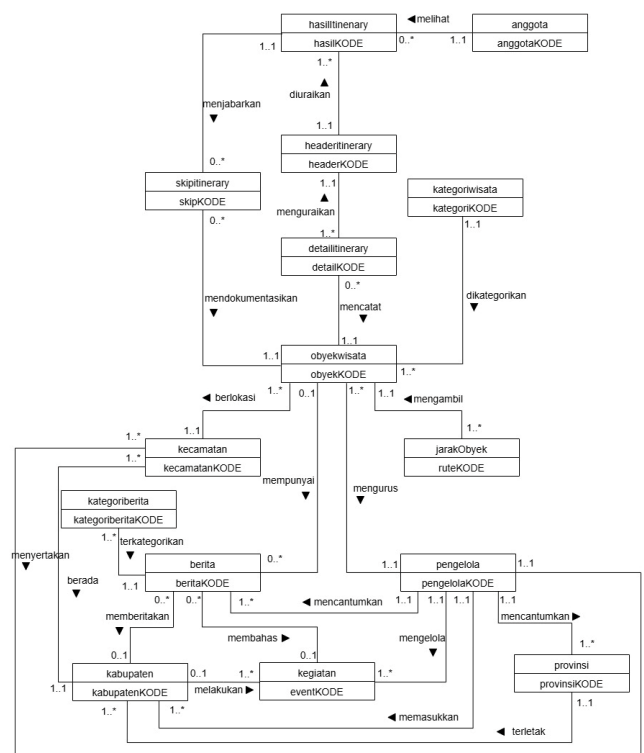
Dalam mengelola data yang sudah ada, selain bisa menambahkan data juga dapat melakukan pengubahan dan penghapusan data yang sudah ada seperti yang dapat dilihat pada **Gambar 5**. Sama seperti tambah destinasi, proses ini dimulai ketika pengguna melakukan *login* dengan akun *admin* dan memilih menu destinasi. Untuk mengubah data destinasi, *admin* harus memilih pilihan edit dan mengisi kembali data yang ingin diubah. Sedangkan untuk menghapus data, *admin* harus memilih pilihan hapus dan mengonfirmasi penghapusan data.



Gambar 5 *Activity Diagram* Edit Destinasi

2.4 Desain Basis Data

Desain basis data adalah tahapan membuat perancangan basis data yang diperlukan. Tahapan ini berfokus untuk mempresentasikan struktur data. Tahapan ini merupakan titik awal untuk membuat sebuah model, yaitu dengan menetapkan definisi serta hubungan antar entitas secara spesifik dengan tujuan untuk memberikan batasan dalam hubungan tersebut [10]. Tujuan dari tahapan ini adalah menentukan entitas, hubungan antar entitas dan attribute yang ada. Desain basis data pada perancangan aplikasi ini akan menggunakan notasi Connolly [11]. Pada tahapan ini juga ditentukan Foreign Key pada entitas. Dengan menentukan attribute ini, kita dapat mengurangi kemungkinan pengulangan data pada basis data. Adapun basis data yang digunakan pada aplikasi ini dapat dilihat pada **Gambar 6**.



Gambar 6 Desain Basis Data

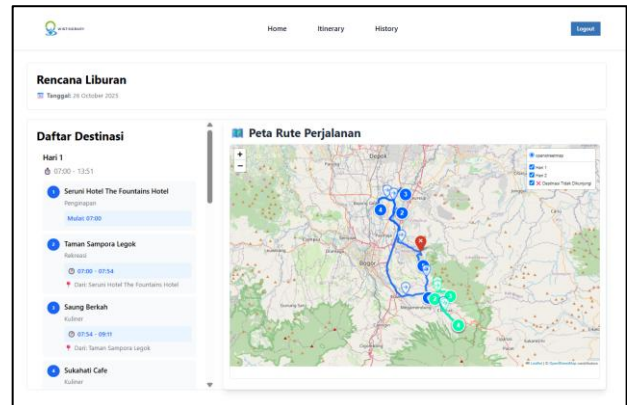
3. Hasil Implementasi dan Pengujian

3.1. Metode Perancangan Aplikasi

Hasil implementasi rancangan sistem adalah dalam bentuk aplikasi. Tampilan pada aplikasi ini terinspirasi dari perancangan sebelumnya yang membahas web pariwisata [12]. **Gambar 7** adalah tampilan *form* rekomendasi rute perjalanan. Tampilan ini adalah fitur utama dalam aplikasi, berupa pembuatan rekomendasi rute perjalanan. Pada halaman ini, pengguna harus menginput beberapa data yang dibutuhkan oleh program. Pengguna harus menginput judul perjalanan, memilih hotel sebagai titik awal dan titik akhir perjalanan, menginput kapasitas hari, menginput jam mulai dan jam selesai serta memilih beberapa destinasi wisata yang ingin dikunjungi. Setelah semua data telah diisi oleh pengguna, data tersebut akan digunakan untuk menghasilkan rute perjalanan efisien berdasarkan waktu dan tingkat popularitas destinasi wisata.

Gambar 7 Form Rute Rekomendasi Perjalanan

Setelah pengguna membuat rute rekomendasi perjalanan, pengguna akan dialihkan ke halaman visualisasi peta yang dapat dilihat pada **Gambar 8**. Tampilan ini adalah hasil dari program yang telah memproses dan membuat urutan kunjungan destinasi wisata secara optimal. Pada bagian kiri halaman, ditampilkan daftar destinasi yang berisi urutan tempat wisata yang dikunjungi. Setiap destinasi yang ada, dilengkapi dengan informasi terkait seperti nama, kategori, serta waktu kunjungan yang telah dijadwalkan. Tampilan ini akan memudahkan pengguna untuk dapat memahami alur wisata yang direkomendasikan dari awal perjalanan hingga akhir. Sementara itu, pada bagian kanan halaman, menampilkan peta interaktif rute perjalanan. Peta ini menggunakan penanda untuk menunjukkan lokasi tiap titik destinasi dan garis antar destinasi sebagai jalur yang harus ditempuh. Wisatawan dapat melihat urutan destinasi yang harus dikunjungi beserta jadwal yang telah diatur sedemikian rupa. Fitur ini membantu menyajikan informasi yang dibutuhkan secara terstruktur dan menjadi alat bantu yang efektif dalam rencana perjalanan.



Gambar 8 Visualisasi Peta

Fitur lain yang dapat digunakan pada aplikasi ini adalah pengelolaan data. Fitur ini hanya dapat diakses jika melakukan *login* dengan akun *admin*. Salah satu pengelolaan data yang dapat dilakukan adalah mengelola data obyek wisata yang dapat dilihat pada **Gambar 9**. Halaman ini merupakan fitur yang berfungsi untuk membuat, membaca, mengubah dan menghapus data yang ada pada basis data aplikasi. Pada bagian atas halaman terdapat tabel yang menampilkan semua daftar obyek wisata beserta informasi terkait seperti kode, nama, kecamatan, kategori, alamat, jam operasional, popularitas, deskripsi dan opsi untuk melakukan ubah atau hapus data. Tabel ini dibuat dengan tujuan agar *admin* dapat dengan mudah meninjau seluruh data yang telah tersimpan. Pada bagian bawah tabel, terdapat *form* untuk menambah data obyek wisata. *Form* ini memiliki data yang harus diinput oleh *admin*. Setelah data diinput, sistem akan memproses dan menyimpannya.

Kode	Nama Obyek	Rekomendasi	Kategori	Alamat	Jam Operasional	Popularitas	Deskripsi	Foto	Aksi
32.01.01.001	Kebun Raya Cibinong	Cibinong	Rekreasi	Jl. Raya Cibinong - Bogor No. 46, Cibinong, Kab. Bogor, Jawa Barat	07:00 - 17:00	2	Kebun Raya Cibinong adalah kawasan hutan yang berfungsi sebagai pusat penelitian, konservasi, dan rekreasi. Kebun ini memiliki berbagai jenis tumbuhan tropis yang dilindungi, serta fasilitas untuk pengunjung.		Ubah Hapus

Gambar 9 Mengelola Data Obyek Wisata

3.2. Hasil Pengujian Aplikasi

Pengujian pada aplikasi ini dilakukan oleh pengembang aplikasi dengan menggunakan metode *Black Box Testing*. Metode ini adalah jenis pengujian aplikasi yang fokus agar sistem aplikasi berfungsi dengan baik melalui *input* dan *output* tanpa memperhatikan kode program [13]. Tujuan dari pengujian ini adalah memastikan setiap fitur yang disajikan dalam aplikasi dapat berfungsi dengan baik dengan memberikan hasil sesuai dengan yang diharapkan. Setiap fungsi utama pada aplikasi diuji dengan memberikan berbagai macam kondisi *input* dan diamati *output* yang diberikan oleh

sistem. Pengujian ini mencakup beberapa fitur utama aplikasi yang dapat dilihat pada **Tabel 1**.

Tabel 1 Hasil Pengujian Aplikasi

Pengujian	Hasil	Status
User membuat <i>itinerary</i> dengan mengisi seluruh <i>field</i> yang dibutuhkan	User berhasil membuat <i>itinerary</i> dan dapat melihat hasilnya pada peta	Berhasil
User membuat <i>itinerary</i> namun tidak mengisi seluruh <i>field</i> yang ada	User gagal membuat <i>itinerary</i> dengan pesan seluruh <i>field</i> harus terisi	Berhasil
User melihat riwayat <i>itinerary</i>	User dapat melihat riwayat <i>itinerary</i> yang pernah dibuat sebelumnya	Berhasil
User melihat destinasi yang dan informasinya pada peta	User dapat melihat informasi mengenai destinasi pada penanda di peta	Berhasil
User melihat daftar destinasi yang tidak dapat dikunjungi selama perjalanan	User dapat melihat daftar destinasi yang tidak dapat dikunjungi	Berhasil
Admin menambahkan data destinasi	Admin berhasil menambahkan data destinasi	Berhasil
Admin menambahkan data destinasi namun terdapat <i>field</i> yang tidak diisi	Data destinasi tidak tersimpan dan muncul pesan <i>error</i> bahwa <i>field</i> harus diisi	Berhasil
Admin menambahkan data destinasi namun tidak sesuai format	Data destinasi tidak tersimpan dan muncul pesan <i>error</i> bahwa <i>field</i> tersebut tidak sesuai format	Berhasil
Admin mengubah data destinasi	Admin dapat mengubah data destinasi yang sudah ada dan menyimpannya	Berhasil
Admin menghapus data destinasi	Data destinasi terhapus	Berhasil
Mencari destinasi	Admin dapat mencari data destinasi	Berhasil

4. Kesimpulan

Berdasarkan hasil perancangan aplikasi ini, dapat diambil beberapa kesimpulan seperti :

1. Aplikasi berbasis *web* pariwisata berhasil dirancang dan diimplementasikan dengan baik.
2. Perancangan aplikasi menggunakan metode pengembangan *Waterfall* dengan masalah perencanaan perjalanan yang dimodelkan dengan *Capacitated Vehicle Routing Problem* berhasil diterapkan dengan baik.
3. Kekurangan aplikasi ini adalah aplikasi yang bersifat lokal. Selain itu adalah data destinasi yang terbatas, hal ini dikarenakan data destinasi wisata tergantung pada data yang diinput oleh

admin. Hal ini menyebabkan kekurangan variasi destinasi wisata yang dapat diberikan pada aplikasi.

4. Untuk pengembangan selanjutnya, aplikasi dapat diperluas agar mendukung berbagai macam destinasi wisata. Sistem juga dapat dikembangkan dengan integrasi API peta *real-time* untuk menambahkan data destinasi yang ada

REFERENSI

- [1] B. S. A. Utami and A. Kafabih, "Sektor pariwisata Indonesia di tengah pandemi COVID 19," *Jdep*, vol. 4, no. 1, pp. 8–14, 2021.
- [2] I. Utama and N. P. D. Krismawintari, "POPULARITAS DAYA TARIK WISATA PANTAI DI BALI SEBELUM DAN MASA PANDEMI COVID-19 MENURUT WISATAWAN DOMESTIK (Popularity of Beach Tourism Attraction in Bali before and Time of the COVID-19 Pandemic by Domestic Tourists)," *Utama, IGBR and Krismawintari, NPD (2022) "POPULARITAS DAYA TARIK WISATA PANTAI DI BALI SEBELUM DAN MASA PANDEMI COVID-19 MENURUT WISATAWAN DOMESTIK"*, *Sebatik*, vol. 26, no. 1, pp. 326–331, 2022.
- [3] V. Wangi, T. Jap, and W. Wasino, *Start to End: Recommended Travel Routes Based on Tourist Preference*, vol. 852. 2020. doi: 10.1088/1757-899X/852/1/012163.
- [4] G. T. S. Lee, D. Arisandi, and Wasino, "Travel App - showing nearest tourism site using Haversine formula and directions with Google Maps," *IOP Conf Ser Mater Sci Eng*, vol. 852, no. 1, p. 012161, 2020, doi: 10.1088/1757-899X/852/1/012161.
- [5] W. Wasino, E. Dewayani, D. Arisandi, and N. Andanwertti, "Informasi Destinasi Wisata Berbasis Peta Google: Studi Kasus Kabupaten Gunungkidul," *JTT (Jurnal Teknologi Terapan)*, vol. 5, no. 2, pp. 86–94, 2019.
- [6] Wasino, D. E. Herwindiati, and I. R. Setyawan, "DESIGNING TOURISM MARKETING TOOLS WITH GEOTARGETING OF IP ADDRESSES," *Journal of Southwest Jiaotong University*, vol. 59, no. 2, 2024, doi: 10.35741/issn.0258-2724.59.2.28.
- [7] A. A. Wahid, "Analisis metode waterfall untuk pengembangan sistem informasi," *J. Ilmu-ilmu Inform. dan Manaj. STMIK*, no. November, vol. 1, no. 1, pp. 1–5, 2020.
- [8] L. Setiyani, "Desain Sistem: Use Case Diagram," in *Prosiding Seminar Nasional Inovasi Dan Adopsi Teknologi (INOTEK)*, 2021, pp. 246–260.
- [9] H. P. Putra, H. Maulana, E. Triandini, and P. F. Nuryananda, "Relasional Desain Activy Diagram Sistem Informasi Agen Travel," in *Prosiding Seminar Nasional Teknologi dan Sistem Informasi*, 2022, pp. 238–241.
- [10] I. R. Mukhlis *et al.*, "Rancangan basis data absensi pegawai menggunakan MySQL dengan conceptual data model (CDM), physical data model (PDM), dan entity relationship diagram (ERD)," *Computing Insight: Journal of Computer Science*, vol. 6, no. 2, 2024.
- [11] T. M. Connolly and C. E. Begg, *Database systems: a practical approach to design, implementation, and management*. Pearson Education, 2005.

- [12] D. E. Herwindiati and H. Maupa, "The Effects of Tourism Web Development on Prospective Travelers by Considering Persuasive and Liking Principles," *International Journal of Social Science and Business*, vol. 6, no. 4, pp. 574–584, 2022.
- [13] G. J. Myers, T. Badgett, T. M. Thomas, and C. Sandler, *The art of software testing*, vol. 2. Wiley Online Library, 2004.

Marcell Budiman, sedang menempuh gelar S1 pada program studi Sistem Informasi pada Fakultas Teknologi Informasi Universitas Tarumanagara.

Wasino, memperoleh gelar S.Kom. dari Universitas Budi Luhur pada tahun 1999. Kemudian, pada tahun 2001 memperoleh gelar M.Kom. di Sekolah Tinggi Teknik Informasi Benarif Indonesia. Selanjutnya, pada tahun 2024 memperoleh gelar Doktor pada bidang Ilmu Manajemen dari Universitas Tarumanagara. Saat ini sebagai Staf Pengajar Program Studi Sistem Informasi Universitas Tarumanagara.

Lely Hiryanto, memperoleh gelar S.T. dari Universitas Tarumanagara tahun 2001, M.Sc. dari Curtin University of Technology tahun 2006, dan Ph.D. dari Curtin University tahun 2022.