

SISTEM PENDUKUNG KEPUTUSAN REKOMENDASI WISATA DI KABUPATEN KARAWANG DENGAN METODE *PROFILE MATCHING*

Mochamad Yudi Sobari, Purwantoro, Agung Susilo Yuda Irawan

Program Studi Informatika, Fakultas Ilmu Komputer, Universitas Singaperbangsa Karawang
Jl. HS. Ronggo Waluyo, Telukjambe Timur, Karawang, Jawa Barat, Indonesia - 41361
mochamad.16141@student.unsika.ac.id

ABSTRAK

Seiring berjalannya waktu, terjadi perkembangan yang signifikan, terutama dalam teknologi dan ilmu pengetahuan di berbagai bidang. Hal ini mendorong hampir seluruh lembaga untuk mengadopsi teknologi berbasis komputer. Perkembangan saat ini menuntut performa yang cepat, efisien, *user-friendly*, akurat, dan aman, terutama dalam hal pemesanan tiket wisata. Sistem pemesanan tiket wisata konvensional memiliki berbagai kekurangan dan menyulitkan pelanggan karena harus datang langsung ke tempat wisata dan mengantri saat melakukan pemesanan. Hal ini memakan banyak waktu dan tenaga serta membuat proses pemesanan tiket menjadi rumit. Tujuan dari penelitian ini adalah menciptakan sistem informasi pemesanan yang tepat, efisien, akurat, dan aman untuk transaksi pemesanan, pembelian, dan pembayaran tiket wisata. Metode yang digunakan untuk mendukung penelitian ini adalah siklus hidup pengembangan sistem atau biasa disebut dengan SDLC, yakni pendekatan metodologis klasik yang digunakan untuk mengembangkan, merawat, dan memanfaatkan sistem informasi. Metode uji *blackbox* diterapkan untuk melakukan uji sistem, yang fokus pada pengujian fungsionalitas program. Hasil dari penelitian ini menyimpulkan bahwa sistem informasi berbasis *web* untuk pemesanan tiket bertujuan mempermudah akses tiket wisata di Kabupaten Karawang.

Kata kunci: Sistem informasi tiket, pemesanan tiket, SDLC

1. PENDAHULUAN

Pertumbuhan teknologi informasi saat ini mengalami kemajuan signifikan di berbagai bidang kehidupan. Progres yang pesat dalam teknologi informasi telah memberikan kenyamanan kepada masyarakat dalam mengakses informasi menggunakan perangkat keras maupun perangkat lunak, dimana saja dan kapan saja. Pemanfaatan teknologi informasi ini memiliki peran penting dalam industri pariwisata. Industri pariwisata dianggap sebagai sektor yang dapat memberikan kontribusi besar terhadap devisa daerah dan negara.

Pengembangan situs web dalam sektor pariwisata menjadi suatu aspek yang esensial dan memiliki potensi untuk ditingkatkan melalui penerapan teknologi *MERN stack* sebagai aplikasi halaman tunggal (*single page application*) serta menggunakan perpustakaan JavaScript, dengan tujuan memastikan situs web beroperasi secara waktu nyata. Penambahan fitur sistem berbagi, visualisasi data waktu nyata, dan layanan pesan dalam waktu nyata memiliki potensi untuk memfasilitasi interaksi dan pertukaran informasi antara pengguna dengan kelancaran. Terutama di wilayah Kabupaten Karawang, penerapan sistem informasi pariwisata berbasis web memiliki signifikansi sebagai alat utama dalam menyajikan informasi mengenai lokasi dan destinasi wisata yang terdapat di daerah tersebut. Di masa kini, beragam jenis kegiatan pariwisata didukung oleh berbagai fasilitas dan layanan yang diberikan oleh berbagai pihak, termasuk masyarakat, pelaku usaha, pemerintah, dan pemerintah daerah, sebagaimana

dinyatakan dalam Pasal 1 angka 4 dari Undang-Undang No. 10 Tahun 2009.

Proses promosi pariwisata yang masih kurang tersebar secara luas menjadi masalah tersendiri bagi masyarakat yang ingin berkunjung. Hal ini berdampak pada keterbatasan pengetahuan mengenai lokasi wisata, opsi penginapan di sekitar area pariwisata, biaya tiket masuk, dan pesona wisata yang ditawarkan. Oleh karena itu, penulis merancang sistem pendukung keputusan rekomendasi wisata di Kabupaten Karawang dengan metode *Profile Matching*, penelitian ini akan mengupas lebih dalam tentang implementasi teknologi ini dalam memfasilitasi pemilihan destinasi wisata yang sesuai dengan preferensi dan profil pengunjung sehingga memudahkan masyarakat dalam memilih destinasi wisata sesuai dengan keinginannya.

2. TINJAUAN PUSTAKA

2.1. SPK (Sistem Pendukung Keputusan)

Menurut penelitian yang dilakukan oleh Setyaningsih [1], terdapat berbagai definisi mengenai Sistem Pendukung Keputusan (SPK), yaitu sebagai berikut:

1. Metode Scott mengungkapkan bahwa Sistem Pendukung Keputusan (SPK) ialah suatu sistem komputer berinteraksi yang membantu para pengambil keputusan dalam memecahkan situasi masalah yang bersifat setengah terstruktur atau tidak terstruktur, melalui pemanfaatan data dan model keputusan. Prinsipnya adalah untuk meningkatkan efisiensi dalam proses pengambilan keputusan.

2. Alavi dan Napier menjelaskan bahwa Sistem Pendukung Keputusan (SPK) merupakan rangkaian prosedur pengolahan data dan informasi dengan memanfaatkan model guna menghasilkan berbagai solusi, bertujuan untuk memberikan bantuan kepada manajemen dalam proses pengambilan keputusan. SPK perlu memiliki sifat yang simpel, *user-friendly*, dan mampu beradaptasi.
3. Menurut pendapat Little, Sistem Pendukung Keputusan (SPK) adalah suatu sistem informasi yang beroperasi melalui komputer dan menghasilkan opsi keputusan sebagai dukungan bagi manajemen dalam mengatasi permasalahan yang bersifat setengah terstruktur atau tidak terstruktur. Pendekatan ini dilakukan dengan memanfaatkan data dan model.
4. Pandangan dari Sparague dan Carlson mengindikasikan bahwa Sistem Pendukung Keputusan (SPK) adalah sistem berbasis komputer yang memberikan dukungan bagi proses pengambilan keputusan terhadap situasi yang memiliki tingkat struktur yang beragam, dari setengah terstruktur hingga tidak terstruktur. Sistem ini menggunakan data dan model, walaupun tidak mengambil alih peran dari pengambil keputusan itu sendiri.
5. Pandangan Al-Hamdany, Sistem Pendukung Keputusan (SPK) merujuk pada sistem informasi yang interaktif, dirancang untuk mendukung proses pembuatan keputusan dengan menyediakan informasi yang spesifik untuk pendekatan penyelesaian masalah dan keperluan aplikasi para pengambil keputusan. Namun, penting dicatat bahwa sistem ini tidak mengambil keputusan atas nama pengguna.

Dari beragam pengertian yang telah disebutkan, dapat ditarik kesimpulan bahwa Sistem Pendukung Keputusan (SPK) merupakan sistem informasi yang memiliki tujuan khusus, yaitu memberikan bantuan kepada manajemen dalam melakukan pengambilan keputusan yang terkait dengan situasi setengah terstruktur. Meskipun dilakukan secara efisien dan efektif, SPK tetap mempertahankan prinsip bahwa ia tidak menggantikan peran pengambil keputusan dalam proses pengambilan keputusan tersebut.

2.2. Profile Matching

Profile Matching adalah langkah perbandingan antara ketrampilan pribadi dengan ketrampilan yang diperlukan untuk suatu posisi tertentu. Maksudnya adalah untuk mengidentifikasi perbedaan dalam ketrampilan, dikenal sebagai "gap", antara kompetensi individu dan kebutuhan. Semakin sempit gap yang timbul, semakin tinggi penilaian ketrampilan individu, yang menandakan bahwa kesempatan karyawan untuk mengisi posisi tersebut lebih besar. Sistem kompetensi akan mencerminkan prestasi yang dicapai dan potensi sumber daya

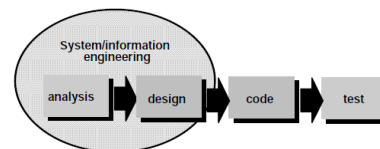
manusia yang sesuai dengan unit kerjanya [2]. Penggunaan *Profile Matching* dapat dilihat apakah kompetensi karyawan telah sesuai dengan tugas pekerjaan yang diemban olehnya. Hal ini membantu dalam menilai kinerja karyawan dan mengidentifikasi potensi yang dimiliki oleh individu yang dapat diaplikasikan dalam tugas pekerjaan yang dimilikinya [2].

2.3. Software Development Life Cycle (SDLC)

SDLC atau biasa disebut dengan siklus hidup pengembangan sistem adalah suatu rangkaian tahapan dalam menciptakan dan mengubah sistem, juga mencakup model dan pendekatan yang diterapkan untuk mengembangkan sistem tersebut [3]. SDLC ini telah menjadi pijakan bagi berbagai Upaya pengembangan sistem informasi, menghasilkan kerangka kerja yang digunakan untuk merencanakan dan mengawasi berbagai tahapan dalam proses pembangunan sistem informasi. Dalam proses pengembangan sistem informasi, pemilihan metode SDLC yang tepat memegang peranan penting dalam membentuk kerangka kerja yang sesuai dengan visi dan rencana pengembang. Keputusan untuk menggunakan model SDLC tertentu akan berdampak langsung pada kualitas akhir dari sistem yang dikembangkan, termasuk faktor biaya dan kebutuhan lainnya yang terlibat dalam seluruh proses pengembangan sistem tersebut [4].

2.4. Model Waterfall

Model *Waterfall*, juga dikenal sebagai Model Air Terjun, adalah suatu metode yang berjalan secara sistematis dan berurutan dalam proses pengembangan perangkat lunak. Proses pengembangan dimulai dengan menentukan keperluan pengguna dan melanjutkan melalui tahapan evaluasi keperluan, perancangan atau pemodelan, pengkodean, serta pengujian sistem perangkat lunak. Setelah itu, langkah kontinu pendampingan terhadap perangkat lunak yang telah dibuat diimplementasikan [5].



Gambar 1. Model Air Terjun (*Waterfall*) [5]

Ilustrasi tersebut mengilustrasikan tahapan generik dari model proses *Waterfall*. Nama "*Waterfall*" dipilih karena setiap fase perlu menanti penyelesaian fase sebelumnya dan dilakukan secara berurutan. Berikut adalah rincian penjelasan tentang tiap langkah dalam model *Waterfall*:

1. Tahap analisis, pada tahap ini, dilakukan analisis terhadap spesifikasi kebutuhan sistem, yaitu bagaimana informasi akan ditampilkan dalam

- aplikasi agar dapat diimplementasikan dengan baik ke dalam aplikasi yang akan dibangun.
2. Tahap desain, ada tahap ini, sistem dirancang berdasarkan analisis yang telah dilakukan pada tahap sebelumnya, termasuk merancang model aplikasi dengan mempertimbangkan aspek-aspek secara menyeluruh dan menarik.
 3. Tahap implementasi (*code*), tahap ini melibatkan implementasi dari rancangan sistem yang telah dibuat pada tahap sebelumnya ke dalam bentuk kode atau program.
 4. Langkah pengujian (*testing*), dalam fase ini, sistem yang telah dirancang akan diuji dengan tujuan memastikan bahwa sistem yang dihasilkan sesuai dengan keperluan dan berjalan dengan baik.

2.5. MERN Stack (MySQL, ExpressJS, ReactJS, NodeJS)

Proses pembuatan website umumnya melibatkan integrasi beberapa teknologi yang disatukan menjadi satu kesatuan. Penggabungan teknologi ini sering disebut sebagai "*stack*". Sebagai contoh, kita dapat menyebut tumpukan teknologi LAMP yang melibatkan Linux, Apache, MySQL, dan PHP, dimana semua komponennya adalah perangkat lunak *open-source* [6]. Seiring dengan perkembangan web yang semakin matang dan kebutuhan akan interaktivitas yang lebih tinggi, teknologi *Single Page Applications* (SPA) semakin populer. SPA memberikan pengalaman interaktif yang lebih unggul kepada pengguna dengan cara memuat konten secara dinamis tanpa memerlukan pengulangan muatan seluruh laman *web* [6].

2.6. ReactJS

ReactJS merupakan perpustakaan UI (*User Interface*) yang dikembangkan oleh Facebook untuk memudahkan pembuatan komponen UI yang interaktif, *stateful*, dan dapat digunakan kembali. Menurut penelitian oleh Handoyo dkk pada tahun 2019 [6], ReactJS memiliki fitur-fitur unggulan sebagai berikut:

1. *Declarative* (deklaratif), ReactJS memfasilitasi pembuatan UI yang interaktif dengan pendekatan deklaratif. Artinya, dalam mendesain tampilan untuk setiap state dalam aplikasi, kita cukup mendeklarasikan bagaimana tampilan tersebut harus terlihat, dan ReactJS akan mengelola dan melakukan *update* komponen dengan efisien saat terjadi perubahan data, serta menyajikan komponen yang sesuai dengan perubahan tersebut. Pendekatan deklaratif ini memberikan kemudahan dalam proses *debugging* kode.
2. *Component-based* (berdasarkan komponen), ReactJS memanfaatkan konsep pembuatan komponen, di mana setiap komponen dapat mengelola state mereka sendiri. Komponen-komponen tersebut nantinya dapat dimanfaatkan

untuk merancang antarmuka pengguna (UI) yang lebih kompleks. Logika dari setiap komponen dikodekan dalam bahasa JavaScript, sehingga mempermudah pertukaran data melalui berbagai bagian aplikasi.

3. *Learn once, write anywhere* (pelajari sekali, tulis dimana saja), ReactJS dirancang untuk tidak mengasumsikan teknologi *stack* lain yang digunakan dalam pengembangan aplikasi. Keadaan ini mengizinkan pengembang untuk memanfaatkan aspek-aspek ReactJS tanpa perlu mengubah kode yang telah ada sebelumnya. ReactJS juga memiliki kapabilitas untuk merender di sisi server dengan memakai NodeJS dan juga dapat digunakan untuk membangun aplikasi *mobile* dengan React Native.

2.7. NodeJS

NodeJS adalah suatu lingkungan *runtime open-source* serta lintas platform yang dipakai untuk menjalankan sisi *server* dalam aplikasi situs web. NodeJS dibangun dengan bahasa JavaScript dan dapat dijalankan di beraneka sistem operasi. NodeJS mendasarkan diri pada struktur berdasarkan peristiwa (*event-driven architecture*) serta menyertakan antarmuka *input/output* yang tidak menghambat (*non-blocking*), dimaksudkan untuk meningkatkan kinerja dan kemampuan pengembangan aplikasi, terutama pada situs web yang berorientasi *real-time* [6].

2.8. ExpressJS

ExpressJS adalah sebuah *framework server web* untuk NodeJS yang bertujuan untuk menyederhanakan pembuatan API (*Application Programming Interface*) dan mempermudah pengaturan fungsionalitas aplikasi melalui penggunaan *middleware* dan *routing*. *Framework* ini memberikan utilitas tambahan pada objek HTTP di NodeJS, sehingga memudahkan dalam proses rendering tampilan HTML yang dinamis. Dengan menggunakan ExpressJS, pengembang dapat lebih efisien dalam mengelola rute (*routing*) dan fungsionalitas dari aplikasi *web* yang mereka bangun [6].

2.9. MySQL (My Structure Query Language)

MySQL, atau yang dikenal dengan *My Structure Query Language*, merupakan sebuah perangkat lunak sistem manajemen basis data SQL (*Database Management System* atau DBMS). Ciri khususnya, MySQL berfungsi sebagai DBMS *multithread*, *multi-user*, dan tersedia secara gratis dengan lisensi GNU *General Public Licence* (GPL) [7]. Beberapa kelebihan dari MySQL adalah sebagai berikut:

1. Keandalan di beragam sistem operasi: MySQL menunjukkan kinerja yang stabil pada berbagai sistem operasi termasuk Windows, Linux, dan sistem operasi lainnya.

2. Sumber terbuka: MySQL disebarluaskan dalam bentuk sumber terbuka atau gratis berdasarkan lisensi GNU General Public Licence (GPL).
3. Banyak pengguna: MySQL mampu menangani penggunaan oleh banyak pengguna secara simultan tanpa mengalami kendala.
4. Kemampuan pertanyaan yang cepat: MySQL menunjukkan kinerja yang efisien dalam mengolah perintah SQL (*query*), memungkinkan untuk memproses sejumlah besar perintah SQL dalam periode waktu yang sama.
5. Perlindungan data: MySQL menjalankan lapisan keamanan yang kokoh, melibatkan aspek seperti tingkat masker *subnet*, *host name*, serta izin masuk pengguna dengan sistem izin yang terperinci, sambil mengenkripsi penyimpanan *password*.
6. Keluwesan dan cakupan antarmuka yang luas: Tidak hanya bersifat mudah diadaptasi dengan berbagai bahasa pemrograman, MySQL juga menawarkan interface ke berbagai aplikasi dan bahasa pemrograman melalui fungsionalitas API (*Application Programming Interface*).
7. Mendapat dukungan dari banyak komunitas, umumnya tergabung dalam forum yang memungkinkan kolaborasi diskusi serta berbagi pengetahuan seputar MySQL. Sebagai contoh, forum <http://forums.mysql.com/> sering dijadikan tempat berinteraksi.

2.10. Use Case Diagram (Diagram Kasus Penggunaan)

Diagram kasus penggunaan (*use case diagram*) adalah gambaran model atas perilaku sistem informasi yang sedang dirancang. Diagram ini mengilustrasikan cara interaksi antara satu atau lebih pihak yang terlibat dengan sistem informasi yang tengah dikembangkan [8].

2.11. Alpha Testing

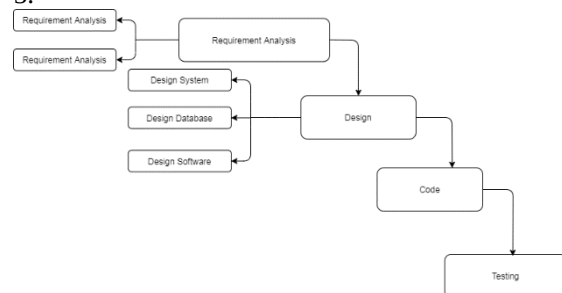
Alpha testing atau pengujian alpha dilaksanakan oleh pihak pengembang atau di lingkungan pengembang sendiri. Sasaran utamanya adalah memverifikasi bahwa aplikasi beroperasi secara tepat sesuai kebutuhan dan sasaran yang telah ditetapkan. Pengujian ini difokuskan pada segi persyaratan fungsional dari perangkat lunak [9].

2.12. Beta Testing

Tahap pengujian beta, yang dikenal sebagai uji beta, adalah proses pengujian yang berfungsi untuk menilai apakah sistem dapat diterima atau perlu direvisi sebelum digunakan secara luas. Uji beta meliputi lima aspek dimensi kualitas, termasuk konten, fungsi, kegunaan, navigabilitas, dan kinerja. Proses pengujian ini memanfaatkan alat berbentuk kuesioner untuk melakukan pengukuran dalam bentuk kuantitatif dan menghasilkan informasi data yang akurat [10].

3. METODE PENELITIAN

Penelitian ini mengadopsi metodologi *System Development Life Cycle* (SDLC) dengan menggunakan model *Waterfall*. Pemilihan model ini sesuai dengan kebutuhan penelitian yang mengharapkan tahapan penggunaan sistem yang relatif *user-friendly*, serta menghasilkan sistem yang mampu meningkatkan pendapatan negara maupun daerah. Metode penelitian ini dituangkan dalam bentuk diagram alir yang dapat dilihat pada Gambar 3.



Gambar 2. Diagram Alir Penelitian

Berdasarkan Gambar 3, tahapan yang dilakukan pada penelitian ini terdiri dari 4 tahapan utama yakni *requirement analysis*, *design*, *code*, dan yang terakhir adalah tahapan *testing*.

4. HASIL DAN PEMBAHASAN

4.1. Requirement Analysis

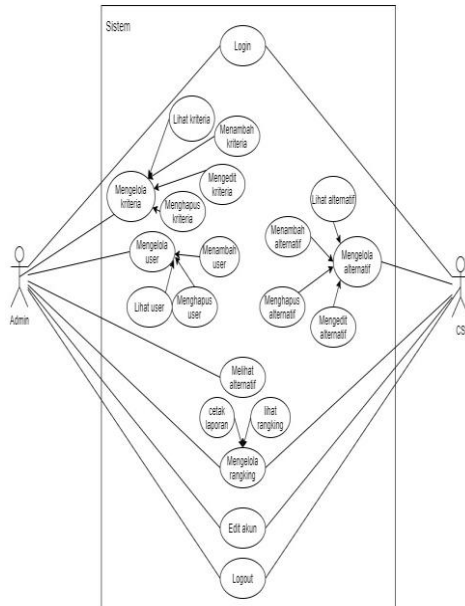
1. Kebutuhan Pengguna, aplikasi yang dibuat memiliki dua pengguna dengan peran yang berbeda, yaitu *Admin* dan *User/Guest*. *Admin* bertanggung jawab untuk mengelola kriteria, mengatur akun pengguna, mengelola hasil perankingan, dan mengelola akun pribadi mereka sendiri. Sementara itu, pengguna dengan peran *User/Guest* bertugas mengelola alternatif, mengakses hasil perankingan, dan mengatur akun pribadi mereka sendiri.
2. Kebutuhan Fungsional, berikut adalah beberapa kebutuhan fungsional pada sistem ini:
 - Admin dapat mengelola akun user, termasuk membuat, menghapus, dan melihat data user.
 - Admin dapat melihat alternatif atau data yang akan digunakan dalam pendukung keputusan.
 - Admin bisa melihat dan mencetak hasil pengurutan yang telah dilakukan.
 - Admin dapat mengelola kriteria, termasuk membuat, mengedit, atau menghapus kriteria dalam sistem.
 - Admin dapat merubah password dan profile akun miliknya.
 - User juga memiliki kemampuan yang serupa, yaitu bisa membuat, mengedit, atau menghapus alternatif, melihat hasil pengurutan, mencetak hasil pengurutan, serta merubah password dan profile akun miliknya.

- Admin memiliki fitur tambahan, yaitu dapat menambahkan nilai bobot kriteria secara dinamis.

4.2. Design

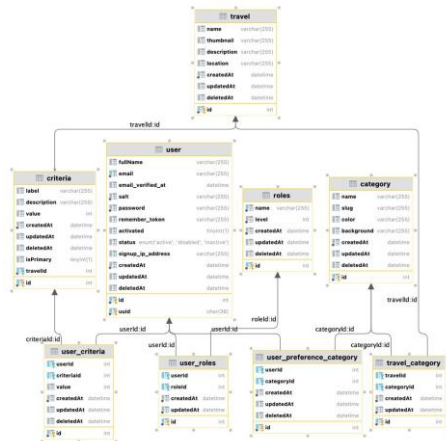
Tahapan ini terdiri dari 3 proses desain yang dijalankan, yaitu:

1. Desain arsitektur sistem pada penelitian ini disajikan dalam bentuk *use case diagram* yang dapat dilihat pada Gambar 4.



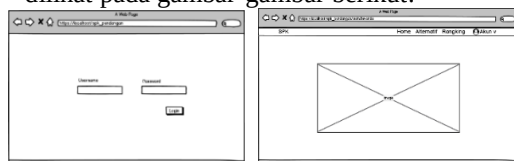
Gambar 3. Use Case Diagram

2. Desain basis data pada penelitian ini dapat dilihat pada Gambar 5.

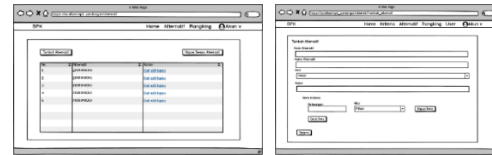


Gambar 4. Desain Basis Data Penelitian

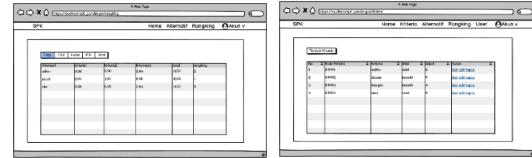
3. Desain antarmuka dari penelitian ini dapat dilihat pada gambar-gambar berikut:



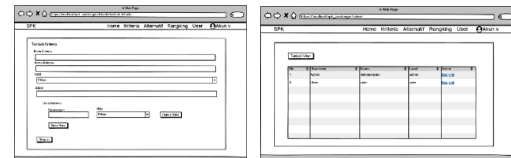
Gambar 5. Halaman Login dan Home



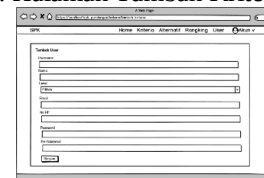
Gambar 6. Halaman Menu Alternatif dan Tambah Alternatif



Gambar 7. Halaman Menu Ranking dan Menu Kriteria



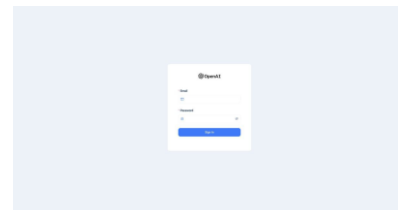
Gambar 8. Halaman Tambah Kriteria dan User



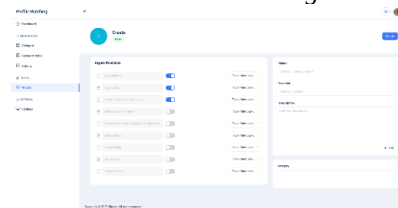
Gambar 9. Halaman Tambah User

4.3. Code

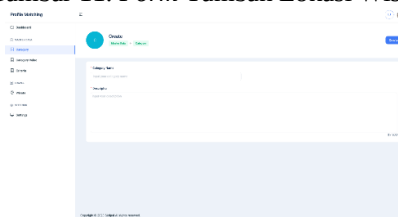
Berikut ini adalah tampilan hasil dari aplikasi sistem pendukung keputusan prioritas produksi undangan menggunakan teknologi MERN Stack dan DBMS MySQL:



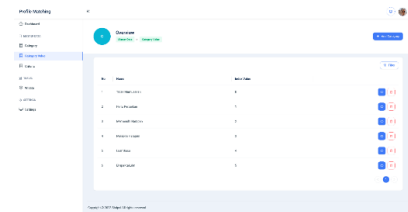
Gambar 10. Form Login



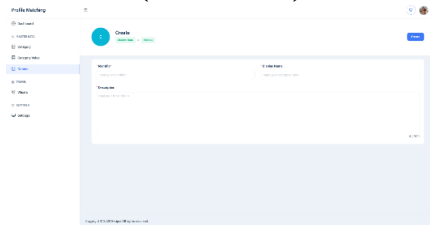
Gambar 11. Form Tambah Lokasi Wisata



Gambar 12. Form Tambah Category Travel



Gambar 13. Form Tambah Category Value (Standar GAP)



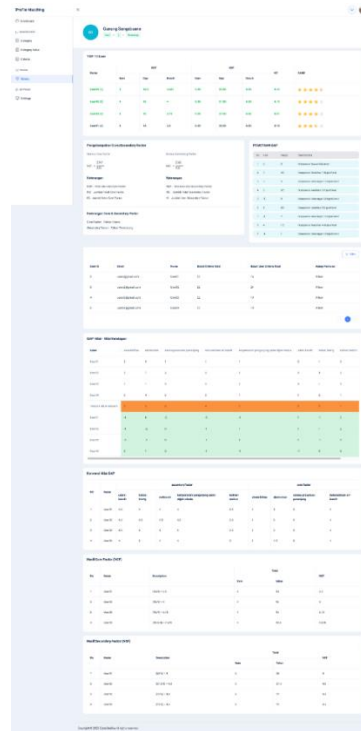
Gambar 14. Form Tambah Criteria

4.4. Testing

Uji aplikasi ini dijalankan untuk menguji setiap fungsi yang ada di dalam aplikasi yang telah dikembangkan, sekaligus untuk mendeteksi potensi kesalahan atau *bug*. Proses pengujian ini melibatkan metode *alpha testing* dan *beta testing*. Berikut adalah laporan hasil pengujian aplikasi:

1. *Alpha Testing*, pengujian ini dilakukan dengan menerapkan metode *black box*, di mana proses pengujian dilakukan tanpa menelusuri kode program internal aplikasi. Sasaran dari tahap ini adalah untuk memverifikasi apakah aplikasi mampu beroperasi dengan baik pada beragam

sistem operasi. Hasil dari uji alpha pada aplikasi sistem pendukung keputusan prioritas produksi undangan adalah sebagai berikut:



Gambar 15. Tampilan Calculate Lokasi Wisata (Profile Matching)

Tabel 1. Alpha Testing

Input	Planning	Response System	Result
Hasil Uji Data Normal			
Login	Masuk dengan hak akses sama	Masuk dengan hak akses sama	✓
Tambah Kriteria (data sesuai)	Muncul notifikasi	Muncul notifikasi	✓
Edit Kriteria (data sesuai)	Muncul notifikasi sukses	Muncul notifikasi sukses	✓
Hapus Kriteria	Muncul konfirmasi aksi dan pilih "ya"	Menghapus kriteria yang dipilih	✓
Tambah Lokasi Wisata (data sesuai)	Muncul notifikasi berhasil	Muncul notifikasi berhasil	✓
Edit Lokasi Wisata (data sesuai)	Muncul notifikasi berhasil	Muncul notifikasi berhasil	✓
Hapus Lokasi Wisata	Muncul konfirmasi aksi dan pilih "ya"	Menghapus kriteria yang dipilih	✓
Profile Matching	Jika ada data maka tampilkan hasil	Menampilkan hasil	✓
Tambah User (data sesuai)	Muncul notifikasi berhasil	Muncul notifikasi berhasil	✓
Hapus User	Muncul konfirmasi aksi dan pilih "ya"	Menghapus kriteria yang dipilih	✓
Hasil Uji Data Tidak Normal			
Login (username salah)	Muncul notifikasi	Muncul notifikasi	✓
Login (password salah)	Muncul notifikasi password salah	Muncul notifikasi password salah	✓
Tambah Kriteria (data tidak lengkap)	Muncul notifikasi isi data lengkap	Muncul notifikasi isi data lengkap	✓
Edit Kriteria (data tidak lengkap)	Muncul notifikasi isi data lengkap	Muncul notifikasi isi data lengkap	✓
Hapus Kriteria	Muncul konfirmasi aksi dan pilih "batal"	Batal penghapusan kriteria	✓
Tambah Lokasi Wisata (data tidak lengkap)	Muncul notifikasi isi data lengkap	Muncul notifikasi isi data lengkap	✓
Edit Lokasi Wisata (data tidak lengkap)	Muncul notifikasi isi data lengkap	Muncul notifikasi isi data lengkap	✓

<i>Input</i>	<i>Planning</i>	<i>Response System</i>	<i>Result</i>
Hapus Lokasi Wisata	Muncul konfirmasi aksi dan pilih “batal”	Batal penghapusan kriteria	✓
<i>Profile Matching</i>	Jika tidak ada data, tampilkan hasil	Menampilkan “tidak ada data alternatif”	✓
Tambah User (data tidak sesuai)	Muncul notifikasi isi data lengkap	Muncul notifikasi isi data lengkap	✓
Hapus User	Muncul konfirmasi aksi dan pilih “batal”	Batal penghapusan kriteria	✓

Dari hasil pengujian alpha yang telah dilakukan, dapat disimpulkan bahwa sistem telah mampu mengeksekusi semua kasus uji sesuai dengan rencana dan harapan yang telah ditetapkan. Hal ini terjadi baik pada uji menggunakan data yang bersifat normal maupun pada uji dengan data yang memiliki karakteristik tidak normal.

2. *Beta Testing*, pengujian ini merujuk kepada tahap pengujian aplikasi yang dilakukan oleh para pengguna langsung. Kuesioner uji beta dijadikan sebagai alat bagi pengguna untuk memberikan evaluasi terhadap aplikasi yang telah dikembangkan. Berdasarkan hasil kuesioner ini, dilakukan penghitungan guna mendapatkan kesimpulan mengenai implementasi aplikasi yang telah dibuat. Kuesioner ini terdiri dari 6 pertanyaan yang memanfaatkan skala Likert dengan nilai antara 1 hingga 4. Di bawah ini adalah rangking penilaian yang diberikan oleh pengguna dengan menggunakan skala Likert untuk masing-masing pertanyaan dalam kuesioner.

Tabel 2. Skala Likert

Tingkat Kepuasan	Skala
Sangat Setuju	4
Setuju	3
Tidak Setuju	2
Sangat Tidak Setuju	1

Berdasarkan hasil data kuesioner, didapat persentase masing-masing jawaban dengan rumus:

$$Y = \frac{\Sigma (N. R)}{\text{Skor Ideal}} \times 100\% \quad (1)$$

Keterangan:

- Y : Nilai presentase kelayakan
 X : Jumlah nilai kategori jawaban dikalikan dengan frekuensi ($\Sigma (N. R)$)
 N : Nilai dari setiap jawaban
 R : Frekuensi
 Skor Ideal : Nilai tertinggi dikalikan dengan jumlah sampel

Tabel 3. Presentase penilaian

No.	Persentase Penilaian	Interpretasi
1.	0% - 19,99%	Sangat Tidak Baik
2.	20% - 39,99%	Kurang Baik
3.	40% - 59,99%	Cukup
4.	60% - 79,99%	Baik
5.	80% - 100%	Sangat Baik

Responden terdiri dari 20 orang yang merupakan mahasiswa IT. Data dari pengujian ini ditampilkan pada Tabel 4.

Tabel 4. Data Pengujian

Nama Responden	Nomor Pertanyaan						Total	Maks
	1	2	3	4	5	6		
Ahmad Jaeni	3	3	3	3	4	4	20	24
Rifky Maulana	4	3	3	4	4	3	21	24
Halimatu	4	4	3	4	4	4	23	24
Ismi Damayanti	3	4	4	3	4	4	22	24
Dwi Suci	4	4	4	4	3	4	23	24
Aul	4	3	4	3	4	4	22	24
Putri Dwi Rahayu	4	4	3	3	4	4	22	24
Rama Amanah	4	4	4	4	4	4	24	24
Deeva Yuzrival Ramadhan	4	4	4	4	4	4	24	24
Sahru Ramadan	4	4	3	4	4	4	23	24
Romlah	4	4	4	4	4	4	24	24
Aditya Bayu Prasetyo	4	4	4	4	4	4	24	24
Dicky Yanda	4	4	3	3	3	4	21	24
Anggi Amelia	4	3	4	4	3	4	22	24
Nurul Maulidy	4	4	4	4	3	3	22	24
CV Code Nusa	3	4	3	4	4	3	21	24
Eli	3	4	4	4	3	4	22	24
Mely	4	3	4	3	4	4	22	24
Fadla	4	4	4	3	4	4	23	24
Ridho Permana	3	4	4	4	3	4	22	24
Jumlah							447	480

Jumlah responden dalam pengujian beta adalah 20 orang. Terdapat enam pertanyaan dalam kuesioner, dan skor maksimal yang dapat diperoleh oleh tiap responden adalah 24 poin. Untuk menghitung persentase kelayakan berdasarkan skor yang diperoleh, kita dapat menggunakan persamaan 1 sebagai berikut:

$$Y = \frac{\sum (N.R)}{\text{Skor Ideal}} \times 100\% = \frac{447}{480} \times 100\% = 93,12\%$$

Hasil perhitungan persentase kelayakan sebesar 93,12%, dapat disimpulkan bahwa aplikasi telah memenuhi standar *usability* dengan kategori "Sangat Baik" berdasarkan tabel persentase kelayakan. Persentase kelayakan yang tinggi menunjukkan bahwa sebagian besar pengguna memberikan penilaian yang sangat positif terhadap aplikasi, dan aplikasi tersebut berfungsi dengan baik dan *user-friendly*. Hal ini menunjukkan bahwa pengujian beta telah berhasil dan aplikasi siap untuk diluncurkan secara resmi untuk digunakan oleh pengguna.

5. KESIMPULAN DAN SARAN

Berlandaskan pada kesimpulan yang diperoleh dari sistem informasi pemesanan tiket wisata berbasis *online*, dapat dinyatakan bahwa sistem ini berperan sebagai alat bantu pengambilan keputusan yang membantu masyarakat dalam memilih tujuan wisata yang sesuai dengan keinginan dan kebutuhan mereka. Selain itu, aplikasi ini juga mampu membantu pengguna untuk mengetahui seberapa cocok suatu lokasi wisata dengan minat dan keinginan mereka, sehingga pengunjung dapat memilih lokasi yang paling sesuai untuk dikunjungi. Adanya sistem pendukung keputusan ini, proses pemilihan lokasi wisata menjadi lebih efisien dan tepat, sehingga pengguna dapat merencanakan perjalanan wisata mereka dengan lebih baik dan memaksimalkan pengalaman wisata yang menyenangkan. Penelitian selanjutnya dapat mengganti *database* yang digunakan menjadi *firebase*, dan dapat mengembangkan keamanan pada *database* sistem.

DAFTAR PUSTAKA

- [1] W. Setyaningsih, *Konsep Sistem Pendukung Keputusan*. Malang: Yayasan Edelweis, 2015.
- [2] A. Muqtadir and I. Purdianto, "Sistem Pendukung Keputusan Kenaikan Jabatan Menggunakan Metode Profile Matching (Studi Kasus di PT. Industri Kemasan Semen Gresik)," in *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*, Jun. 2013, pp. 48–55.
- [3] R. Ingg et al., "Penerapan System Development Life Cycle (SDLC) dalam Mengembangkan Framework Audio Forensik," *semanTIK*, vol. 4, no. 2, pp. 193–200, 2018, doi: 10.5281/zenodo.2528444.
- [4] A. A. Wahid, "Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi," *Jurnal Ilmu-ilmu Informatika dan Manajemen STMIK*, pp. 1–5, Oct. 2020.
- [5] E. N. A. Romadhoni, T. Widiyaningtyas, and U. Pujiyanto, "Implementasi Model Waterfall pada Pengembangan," in *Seminar Nasional Sistem Informasi Indonesia*, 2015, p. 446.
- [6] R. Handoyo, L. W. Santoso, and A. Setiawan, "Real-Time BPMN Website Menggunakan Teknologi MERN Stack," *INFRA*, vol. 7, no. 2, 2019, [Online]. Available: <https://www.mongodb.com/blog/post/the-modern->
- [7] Anhar, *Panduan Menguasai PHP dan MySQL Secara Otodidak*. Jakarta: Mediakita, 2010.
- [8] R. A.S. and M. Shalahuddin, *Rekayasa Perangkat Lunak Tersruktur dan Berorientasi Objek*. Bandung: Informatika Bandung, 2019.
- [9] D. Tresna and A. S. Milah, "Pengembangan Aplikasi Komik Hadis Berbasis Android," *Jurnal Algoritma*, 2017.
- [10] W. Usino and G. D. P. Utama, "E-CRM Dengan Metodologi Fast (Framework For The Application of System Technique) Sebagai Upaya Peningkatan Kepuasan dan Loyalitas Pelanggan: Studi Kasus UKM U-ME Online," *TELEMATIKA MKOM*, p. 36, 2018.