

BAB 3

METODOLOGI PENELITIAN

3.1 Metodologi

Berikut ini merupakan metodologi dari penelitian yang akan dilaksanakan. Metodologi terbagi menjadi lima tahap, yaitu perencanaan, desain, implementasi, pengujian, dan evaluasi.

1. Perencanaan dan Pengumpulan *Requirement*

Sebelum melakukan rancang bangun sistem, sebaiknya ditentukan dahulu konsep, fitur-fitur yang akan digunakan, serta target demografi atau user utama dari sistem. Untuk kasus ini, konsep yang digunakan adalah aplikasi *cross-platform* yang bisa dibangun dalam bentuk *website* maupun aplikasi *mobile*, dengan fokus utama pembangunan *website*. Target demografi dari *website* ini merupakan wisatawan yang berencana untuk menginap di wilayah Jakarta Barat dan cenderung sulit memilih keputusan, seperti mencari hotel. Selain itu, dilakukan pengumpulan *requirement*, yaitu data-data tentang hotel di Jakarta Barat dan kriteria yang bersangkutan.

2. Desain

Setelah melakukan perencanaan aplikasi, tahapan desain dimulai. Tahapan ini merupakan tahap untuk membangun kerangka dari aplikasi. Tahapan ini dibagi menjadi tiga, yaitu desain aplikasi, desain antarmuka, dan desain basis data.

(a) Desain aplikasi

Desain aplikasi adalah tahapan untuk menentukan alur dari aplikasi. Desain aplikasi menggunakan diagram seperti *Data Flow Diagram* (DFD) dan *flowchart* untuk menentukan peran user dan cara kerja dari aplikasi.

(b) Desain antarmuka

Desain antarmuka adalah tahapan untuk membuat UI (*User Interface*) atau tampilan antarmuka dari aplikasi. Desain antarmuka menggunakan *website* Figma untuk memudahkan proses pembuatan tampilan

antarmuka. Proses ini juga berperan dalam pembuatan prototipe dan *mockup*.

(c) Desain basis data

Desain basis data adalah proses untuk menentukan tabel yang akan ada di basis data aplikasi ini. Proses untuk membuat basis data dimulai dari pengumpulan *requirement*, lalu membangun *Entity Relational Diagram* (ERD) untuk menentukan atribut dan relasi antar atribut yang dibutuhkan dalam proyek ini.

3. Implementasi

Setelah melakukan desain, tahap implementasi dimulai. Tahap implementasi adalah tahap realisasi dari *website*, berdasarkan desain aplikasi, UI, dan basis data yang telah ditentukan sebelumnya. Di tahap ini akan dilakukan pemrograman dari *website* dan sistem rekomendasi. Pemrograman akan dilakukan di Visual Studio Code sebagai IDE (*Integrated Development Environment*) utamanya, menggunakan Ionic/React untuk membangun *front end* dan PHP murni untuk membangun *back end*. Output yang dihasilkan dari implementasi adalah sebuah aplikasi yang bisa dijalankan di *browser* yang *responsive* baik di komputer maupun telepon genggam (*mobile*).

4. Pengujian

Setelah sistem rekomendasi selesai dibuat di tahap implementasi, tahap pengujian dimulai. Beberapa penguji coba akan terpilih untuk menguji aplikasi ini dengan melakukan navigasi dan menggunakan fitur yang tersedia, yaitu mengisi kriteria yang diinginkan dan mendapatkan rekomendasi hotel. Tahap uji coba menggunakan metode *black box testing*.

5. Evaluasi

Tahap terakhir dalam penelitian setelah tahap uji coba adalah tahap evaluasi. Di sini, tingkat kepuasan user setelah menggunakan sistem rekomendasi akan diukur dengan menyebarkan kuesioner yang menggunakan metode EUCS dan skala Likert. Responden akan diarahkan untuk menggunakan *website* dan mengisi kuesioner sesuai dengan pengalaman penggunaan *website*, dan hasil survei akan dianalisis untuk menentukan tingkat kepuasan.

3.2 Spesifikasi Sistem

Sebagai penunjang penelitian ini, ada *hardware* dan *software* yang memiliki peran penting dalam pengerjaan penelitian ini. Adapun *hardware* utama yang digunakan adalah sebuah laptop bermerek Asus dengan spesifikasi sebagai berikut:

1. Sistem Operasi Windows 10 Home
2. RAM 8 GB
3. CPU AMD Ryzen 5 3500U
4. GPU Radeon Vega Mobile Gfx 2.10 GHz
5. Ruang Penyimpanan 500 GB

Daftar *software* dan *website* yang digunakan bersama dengan fungsinya adalah sebagai berikut:

1. Mozilla Firefox sebagai *browser* utama untuk mengakses website yang digunakan untuk penelitian ini
2. *Website* diagrams.net sebagai alat untuk desain aplikasi dan struktur basis data
3. *Website* Figma sebagai alat untuk desain antarmuka
4. Visual Studio Code sebagai IDE untuk melakukan pemrograman berbasis *website* dengan bahasa JavaScript dan framework Ionic/React
5. XAMPP untuk mengakses *database* MySQL

3.3 Kebutuhan sistem

Tabel 3.1 merupakan tabel yang menggambarkan kebutuhan sistem atau *system requirements*. Kebutuhan ini didapatkan dari menganalisis penelitian sistem rekomendasi yang pernah ada dan menggunakan komponen yang sering muncul.

Tabel 3.1. Tabel kebutuhan user

No.	Kebutuhan	Deskripsi
1	Login	User dapat login menggunakan email dan <i>password</i> sebagai admin atau user biasa.
2	Daftar Hotel	User dapat melihat daftar hotel dan detil masing-masing hotel. Admin dapat menambah, mengubah data, atau menghapus hotel.
3	Ulasan Hotel	User dapat memberi ulasan berupa rating dan komentar, ataupun memberi favorit terhadap hotel tertentu.
4	Daftar Destinasi	User dapat melihat daftar destinasi (bandara, tujuan wisata, dll). Admin dapat menambah destinasi baru.
5	Rekomendasi Hotel	User menginput prioritas dan preferensi kriteria, dan sistem akan melakukan perhitungan AHP dan WPM berdasarkan input user untuk mendapatkan hotel.
6	<i>History</i>	User dapat melihat <i>history</i> rekomendasi yang telah dilakukan oleh user tersebut. Admin dapat melihat semua <i>history</i> rekomendasi.
7	Profil	User dapat melihat dan mengedit profil (mengganti <i>username</i> , bio, dan foto profil). User dapat melihat daftar ulasan terakhir, hotel favorit, dan history rekomendasi oleh user itu sendiri.
8	<i>Responsiveness</i>	<i>Website</i> harus bersifat <i>responsive</i> dan disesuaikan untuk tampilan komputer dan <i>mobile</i> .
9	<i>Database</i>	<i>Website</i> harus menyimpan data menggunakan <i>database</i> .
10	Tutorial	<i>Website</i> memiliki menu untuk menjelaskan cara menginput data untuk proses rekomendasi.
11	Kriteria	Kriteria yang digunakan dalam perhitungan rekomendasi hotel adalah harga, kelas hotel, adanya Wi-Fi, restoran, dan kolam renang, serta jarak ke destinasi yang dipilih.

3.4 Rancangan sistem

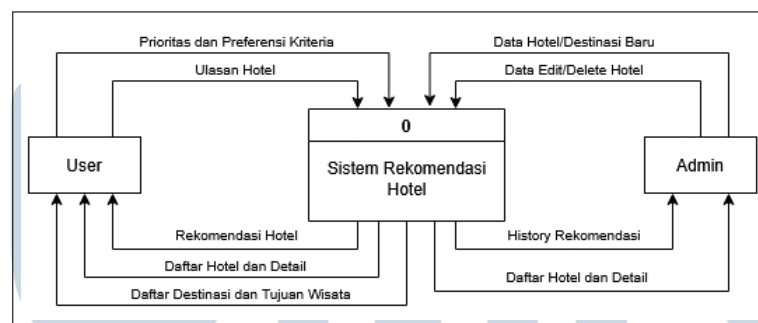
Dalam pembangunan *website* sistem rekomendasi hotel, ada beberapa persiapan yang dilakukan untuk mengerti tampilan dan proses berjalannya *website* serta peran user dalam *website* tersebut. Proses perancangan *website* ini menggunakan tiga jenis diagram, yaitu *Data Flow Diagram*, *flowchart*, dan relasi basis data atau skema, serta *mockup* untuk representasi tampilan *website*.

3.4.1 Data Flow Diagram

Menurut Hartono (2005) dalam Sari dan Siregar (2021), *Data Flow Diagram* atau DFD merupakan diagram yang menggambarkan aliran data dalam suatu sistem menggunakan entitas, proses, dan *data store*. [24] *Data Flow Diagram* untuk *website* sistem rekomendasi hotel adalah sebagai berikut:

1. DFD Level 0

DFD Level 0 merepresentasikan peran-peran utama pada sistem rekomendasi, serta input yang diberikan kepada sistem serta output sistem kepada masing-masing peran.



Gambar 3.1. DFD Level 0

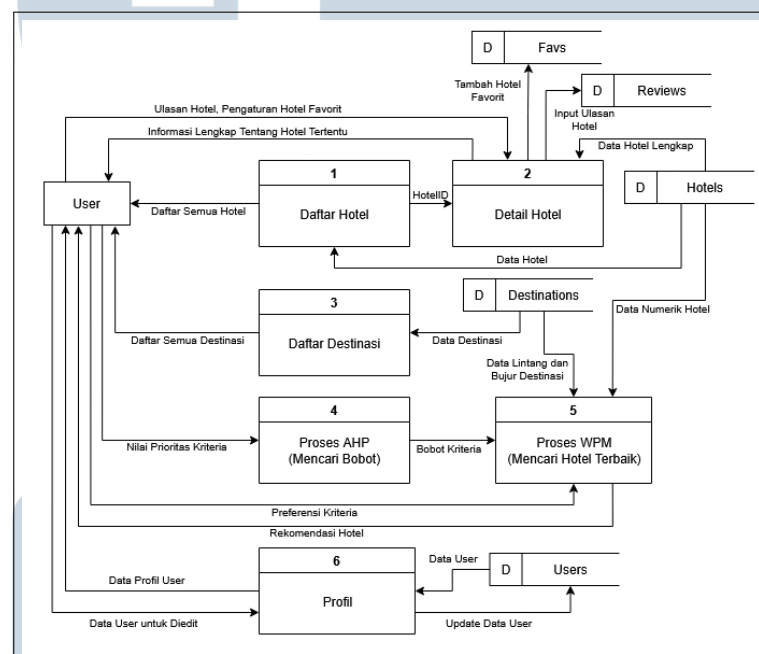
Berdasarkan DFD Level 0 pada Gambar 3.1, ada dua jenis peran utama pada *website*, yaitu user dan admin.

User merupakan pengguna sistem rekomendasi hotel, dengan peran yaitu melihat daftar hotel dan destinasi, mendapatkan rekomendasi hotel berdasarkan preferensi kriteria, serta memberikan ulasan tentang hotel. Untuk keperluan tersebut, sistem mengembalikan data tentang hotel dan/atau destinasi serta hasil rekomendasi hotel.

Admin merupakan pemilik sistem rekomendasi hotel, dengan peran yaitu menambahkan, mengubah data, atau menghapus hotel dan destinasi dan mengelola sejarah atau *history* rekomendasi yang didapatkan oleh user biasa. Untuk keperluan tersebut, sistem mengembalikan daftar *history* serta mengelola proses CRUD di hotel dan destinasi.

2. DFD Level 1

DFD Level 1 merupakan perombakan dari DFD Level 0, yang menjelaskan alur sistem secara detail untuk setiap entitas atau tipe user. DFD Level 1 untuk sistem ini dibagi menjadi dua, yaitu untuk user dan untuk admin.



Gambar 3.2. DFD Level 1 untuk user

Berdasarkan DFD Level 1 pada Gambar 3.2, user memiliki akses terhadap sistem daftar hotel, detail hotel, daftar destinasi, serta kedua proses rekomendasi yaitu menginput perbandingan kriteria (matriks berpasangan) dan preferensi kriteria. Tabel *database* yang digunakan adalah Hotels, Destinations, Reviews, Favorites, dan Users.

Pada menu Daftar Hotel, sistem mengambil daftar hotel dari tabel Hotels dan mengembalikannya ke user. User dapat melihat detail, dan sistem akan mengirim ID Hotel ke *database* untuk mendapatkan detail hotel dan memperlihatkannya di halaman Detail Hotel. User dapat memberi ulasan

Untuk proses perhitungan, pertama user menginput matriks berpasangan ganda pada menu Criteria Priorities yang nantinya akan diproses oleh sistem untuk mendapatkan bobot kriteria menggunakan metode AHP. Bobot ini dikirim ke menu Criteria Preferences, dimana user memilih preferensi kriteria, apakah berupa *benefit* atau *cost* serta destinasi terdekat. Sistem mendapatkan destinasi yang dipilih dari tabel Destinations dan data numerik semua hotel dari tabel Hotels dan melakukan perhitungan WPM untuk mengembalikan hotel terbaik sebagai rekomendasi kepada user.

```

    usecaseDiagram
        actor Admin
        participant U1 as 1  
Daftar Hotel (Admin)
        participant U2 as 2  
Detail Hotel
        participant U3 as 3  
Daftar Destinasi (Admin)
        participant U4 as 4  
History Rekomendasi
        participant U5 as 5  
Profil

        dataStore D1 as D Hotels
        dataStore D2 as D Favas
        dataStore D3 as D Destinations
        dataStore D4 as D History
        dataStore D5 as D Users

        Admin --> U1 : Data Hotel Baru
        Admin --> U1 : Daftar Hotel
        Admin --> U1 : Pengaturan Hotel Favorit
        Admin --> U1 : Data Delete Hotel
        Admin --> U1 : Data Edit Hotel
        Admin --> U2 : HotelID
        Admin --> U3 : Daftar Destinasi
        Admin --> U3 : Data Destinasi Baru
        Admin --> U4 : Daftar History
        Admin --> U5 : Data Profil User
        Admin --> U5 : Data User untuk Diedit

        U1 --> D1 : Input New Hotel
        U1 --> D1 : Edit Hotel
        U1 --> D1 : Delete Hotel
        U1 --> D2 : Data Hotel
        U1 --> D2 : Data Detail Hotel
        U1 --> D3 : Data Destinasi
        U1 --> D4 : Data History
        U1 --> D5 : Data User

        U2 --> D1 : Data Detail Hotel
        U2 --> D2 : Tambah Hotel Favorit

        U3 --> D3 : Input New Destination
        U3 --> D4 : Data History
        U3 --> D5 : Data User

        U4 --> D4 : Data History

        U5 --> D5 : Update Data User
    
```

Berdasarkan DFD Level 1 pada Gambar 3.3, admin memiliki akses terhadap sistem daftar hotel, detail hotel, daftar destinasi, dan *history* rekomendasi. Tabel *database* yang berperan adalah Hotels, Destinations, Favorites, Users, dan History.

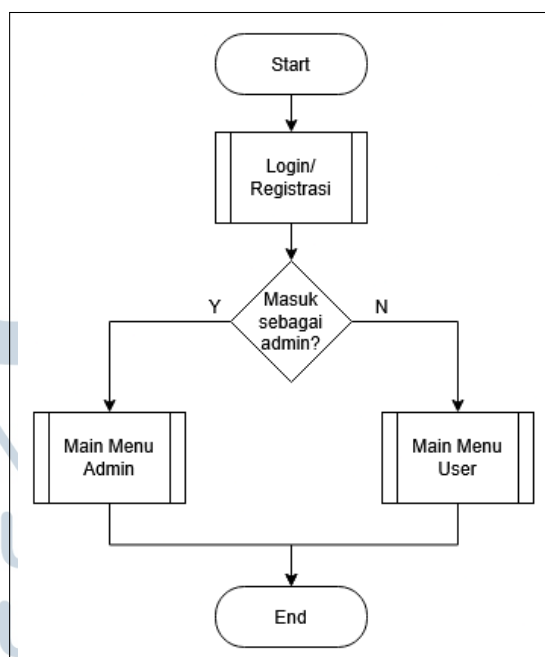
17

melakukan CRUD (*Create, Read, Update, Delete*) terhadap data-data ini. Admin menginput data hotel baru, dan sistem mengirim ke *database* untuk menyimpannya. Untuk proses edit, sistem mengembalikan data hotel untuk diedit nilainya, dan setelah selesai admin mengirim data baru ke sistem untuk disimpan di *database*. Untuk proses *delete*, admin memilih hotel, ID dikirim ke sistem untuk melakukan proses penghapusan hotel. Untuk proses History, sistem mengambil data dari tabel History untuk diperlihatkan di menu History kepada admin. Proses pengaturan hotel sebagai favorit dan penglihatan dan pengeditan profil sama dengan proses untuk user.

3.4.2 Flowchart

Flowchart adalah diagram yang merepresentasikan alur kerja sebuah program menggunakan langkah-langkah yang berurutan dan simbol grafik. [25] *Flowchart* menggambarkan alur sistem dengan mudah dan jelas sehingga mampu dimengerti. Berikut ini merupakan *flowchart* yang dirancang untuk *website* sistem rekomendasi hotel:

1. Flowchart proses utama

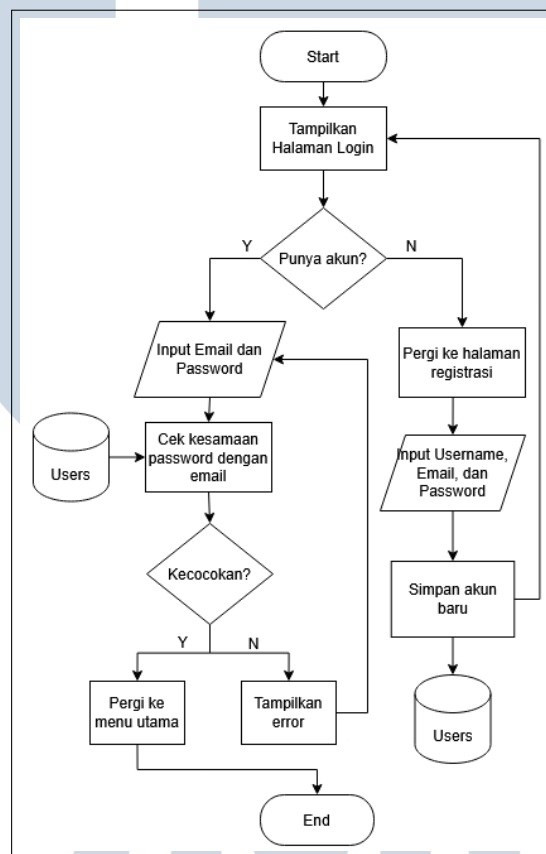


Gambar 3.4. *Flowchart* proses utama

Flowchart utama pada Gambar 3.4 ini merupakan alur sistem secara

umum. Ketika user memasuki *website*, user dapat melakukan login dengan menginput email dan *password*, atau registrasi dengan membuat akun baru. Jika user login sebagai admin, tampilkan menu utama untuk admin, dan jika tidak tampilkan menu utama untuk user.

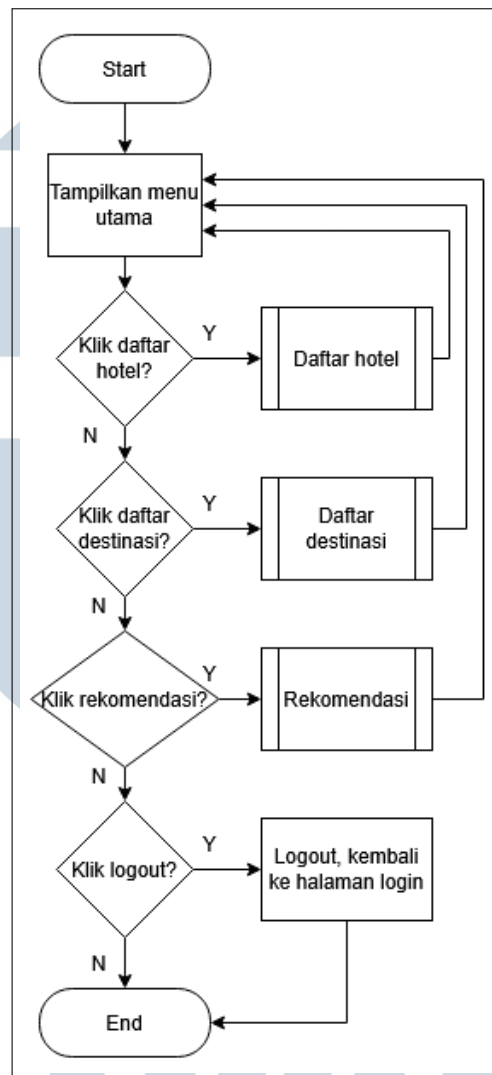
2. Flowchart login dan registrasi



Gambar 3.5. Flowchart login dan registrasi

Flowchart pada Gambar 3.5 menggambarkan proses ketika user berada di halaman login. Jika user belum memiliki akun, user dapat membuat akun baru dengan pergi ke halaman registrasi, menginput email, *username*, dan *password*, dan data user baru akan disimpan di *database*. Jika user memiliki akun, user dapat login dengan memasukkan email dan *password*, dan sistem akan melakukan pengecekan *password* dengan mencocokkan *password* input dengan *password* yang berhubungan dengan email input user. Jika tidak sesuai, maka tampilkan error dan user diminta untuk coba lagi. Jika sesuai, maka pergi ke halaman menu utama.

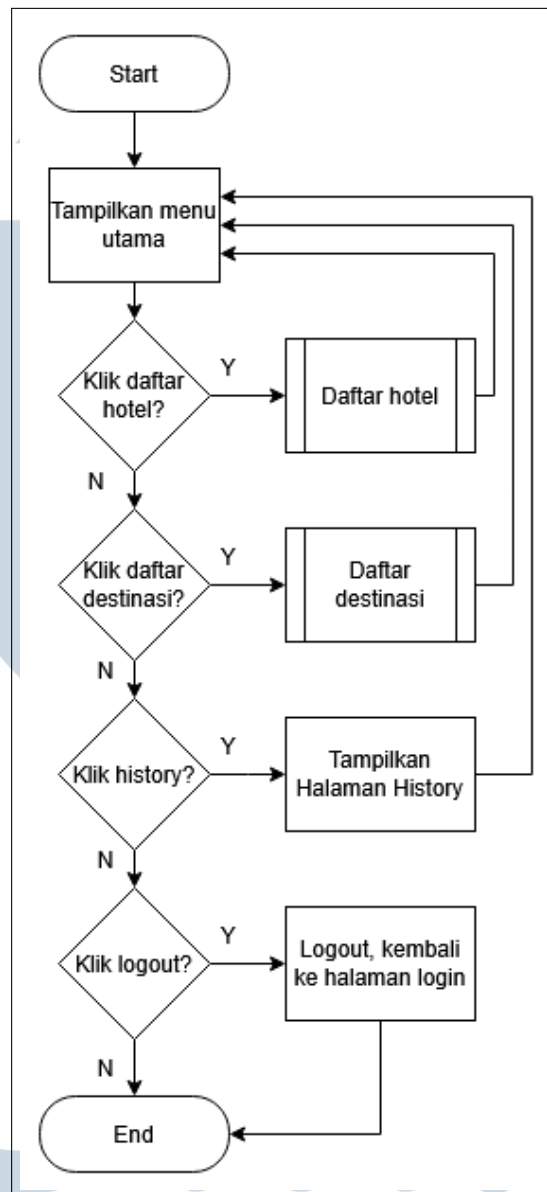
3. Flowchart menu utama untuk user



Gambar 3.6. Flowchart menu utama untuk user

Flowchart pada Gambar 3.6 merupakan alur untuk menu utama ketika melakukan login sebagai user (bukan admin). Ada tiga menu yang bisa diakses oleh user, yaitu daftar hotel, daftar destinasi, dan rekomendasi. Ketika user memilih menu tertentu, maka halaman akan berpindah ke halaman yang bersangkutan.

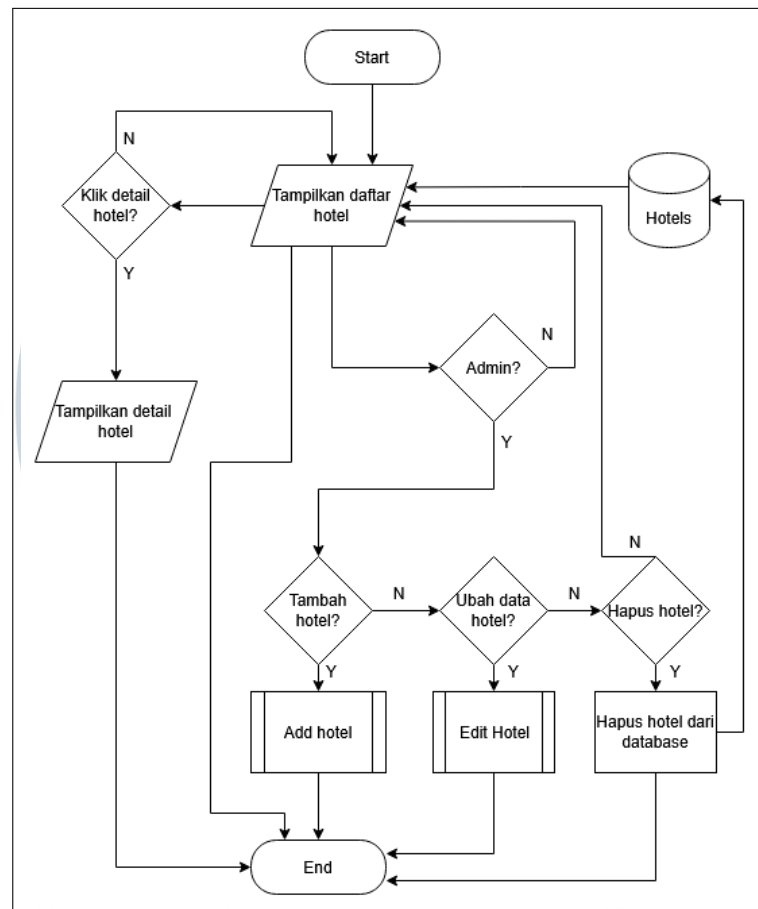
4. Flowchart menu utama untuk admin



Gambar 3.7. Flowchart menu utama untuk admin

Flowchart pada Gambar 3.7 sekilas mirip dengan flowchart menu utama untuk user, bedanya jika user melakukan login sebagai admin maka user dapat melihat *history* atau daftar rekomendasi hotel yang sudah pernah dicatat di *website*.

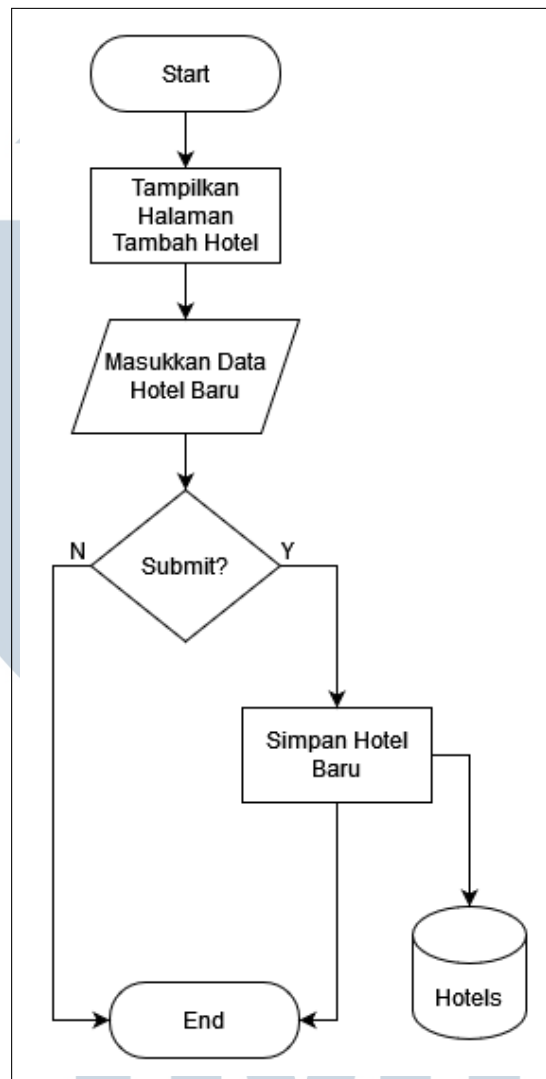
5. Flowchart melihat daftar hotel



Gambar 3.8. Flowchart melihat daftar hotel

Flowchart pada Gambar 3.8 merupakan proses yang ada di menu daftar hotel. User dapat melihat nama dan gambar setiap hotel yang tercatat di *database*, dan juga melihat detail hotel dengan mengklik tanda panah. Admin dapat menambahkan hotel baru, mengubah data hotel, atau menghapus hotel di menu daftar hotel.

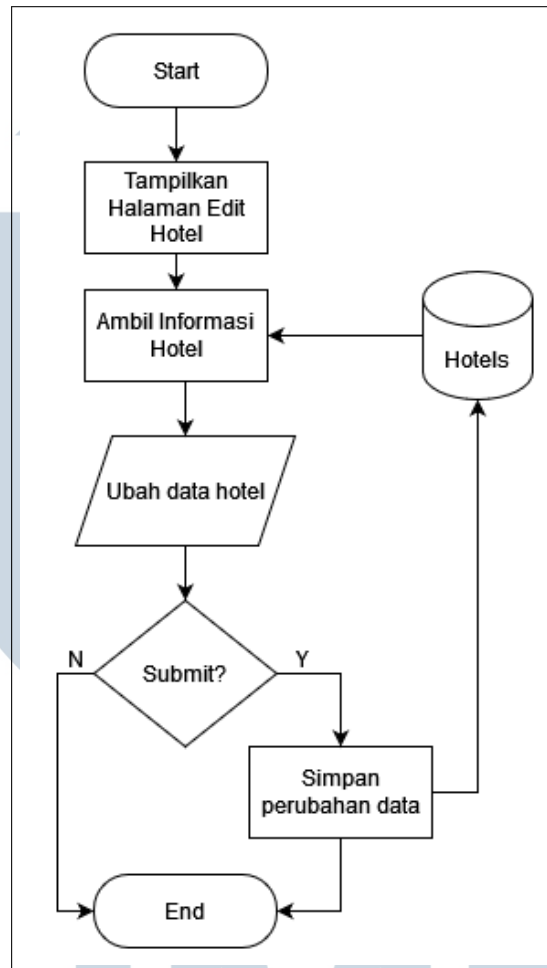
6. Flowchart menambahkan hotel baru



Gambar 3.9. Flowchart menambahkan hotel baru

Flowchart pada Gambar 3.9 adalah flowchart untuk halaman penambahan hotel baru. Halaman ini hanya dapat diakses oleh admin. Ketika memasuki halaman ini, admin dapat menginput data-data yang berkaitan tentang hotel yang akan ditambahkan. Jika menekan tombol *submit*, maka data yang telah diinput disimpan di *database* sebagai objek hotel baru.

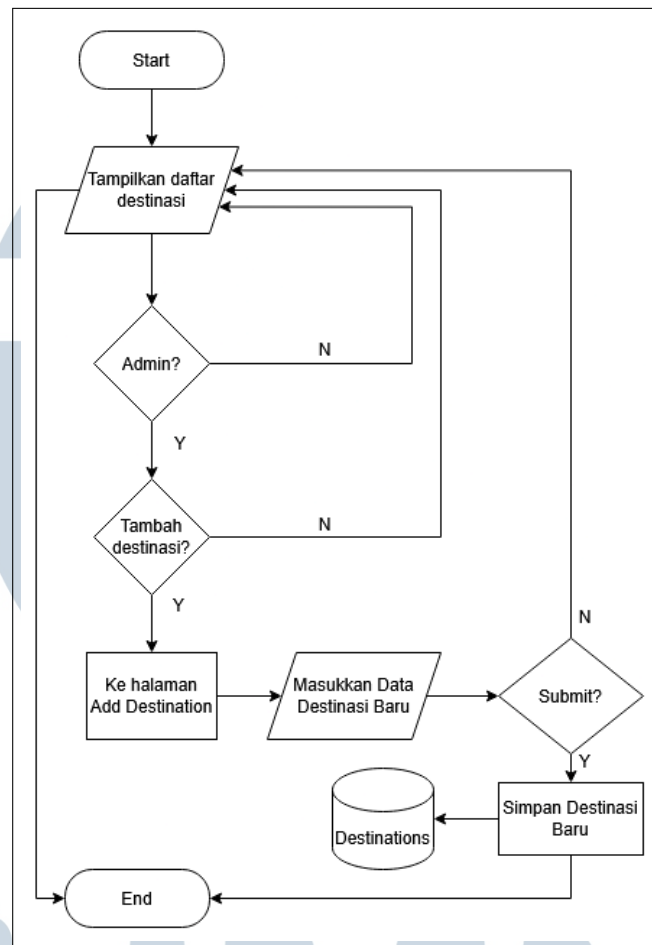
7. Flowchart mengedit data hotel



Gambar 3.10. Flowchart mengedit data hotel

Flowchart pada Gambar 3.10 adalah flowchart untuk halaman perubahan data hotel. Halaman ini hanya bisa diakses oleh admin. Ketika memasuki halaman edit data hotel, sistem akan mengambil data hotel sesuai dengan ID hotel yang dipilih admin dan mengisi formulir dengan data-data yang telah diambil. Admin cukup mengubah data yang perlu diubah. Jika menekan tombol *submit*, maka data hotel yang telah diubah akan tersimpan di *database*.

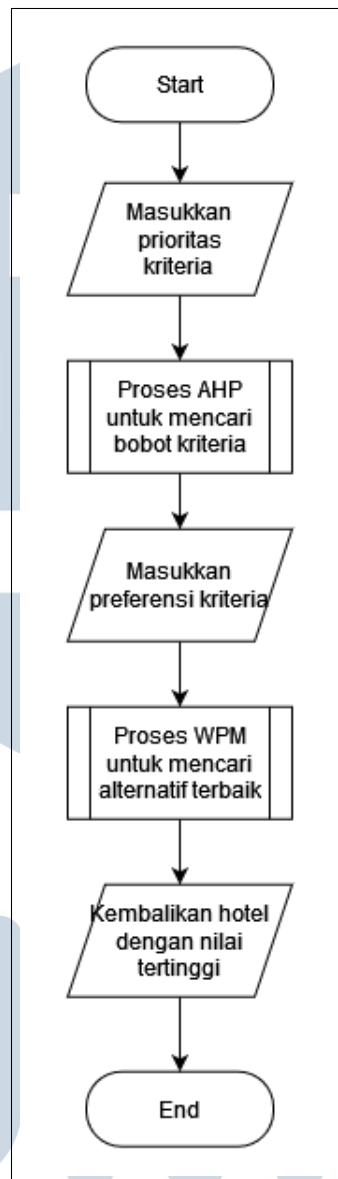
8. Flowchart melihat daftar destinasi



Gambar 3.11. Flowchart melihat daftar destinasi

Flowchart pada Gambar 3.11 merupakan proses yang ada di menu daftar destinasi. User dapat melihat nama dan gambar setiap destinasi yang tercatat di database, dan admin dapat menambahkan destinasi baru di menu Add New Destination.

9. Flowchart rekomendasi hotel

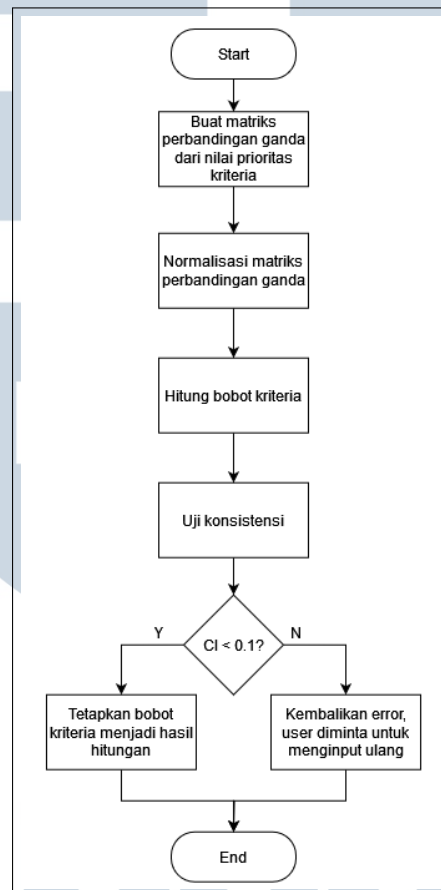


Gambar 3.12. Flowchart rekomendasi hotel

Flowchart pada Gambar 3.12 menggambarkan kerangka utama dari proses rekomendasi hotel. Pertama, user menginput prioritas dari kriteria, yaitu perbandingan kepentingan kriteria menggunakan skala dari 1 sampai 9. Kemudian, dari input tersebut dilakukan proses AHP untuk mencari bobot kriteria. Setelah mendapatkan bobot, user menginput preferensi kriteria, dimana setiap kriteria dapat menjadi kriteria *benefit* atau *cost*. Dengan tipe kriteria dan bobot yang telah didapatkan, sistem akan menggunakan metode

WPM untuk mencari alternatif terbaik, yang nantinya akan dikembalikan ke user sebagai rekomendasi hotel terbaik.

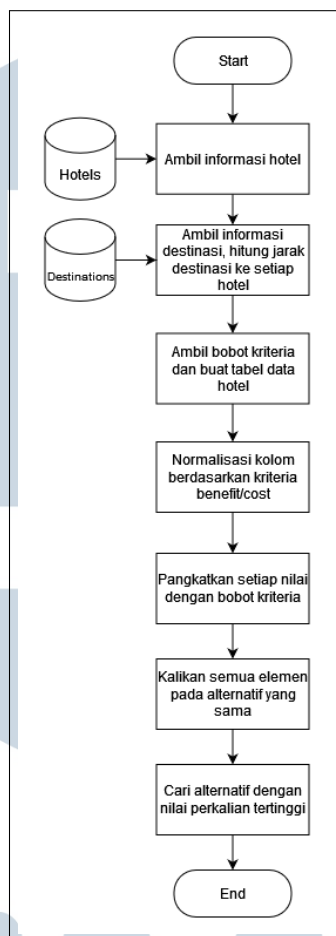
10. *Flowchart* proses AHP untuk menghitung bobot



Gambar 3.13. *Flowchart* proses AHP untuk menghitung bobot

Flowchart pada Gambar 3.13 menggambarkan proses AHP setelah user menginput prioritas kriteria. Dari nilai prioritas setiap kriteria, dibuat matriks berpasangan ganda yang dinormalisasi, kemudian bobot kriteria dihitung menggunakan rumus-rumus AHP. Setelah itu dilakukan uji konsistensi, jika tidak konsisten maka user dituntut untuk mengubah beberapa nilai prioritas, sedangkan jika konsisten maka bobot setiap kriteria ditetapkan dan sistem lanjut ke tahap berikutnya, yaitu menginput preferensi kriteria.

11. Flowchart proses WPM untuk mencari alternatif terbaik

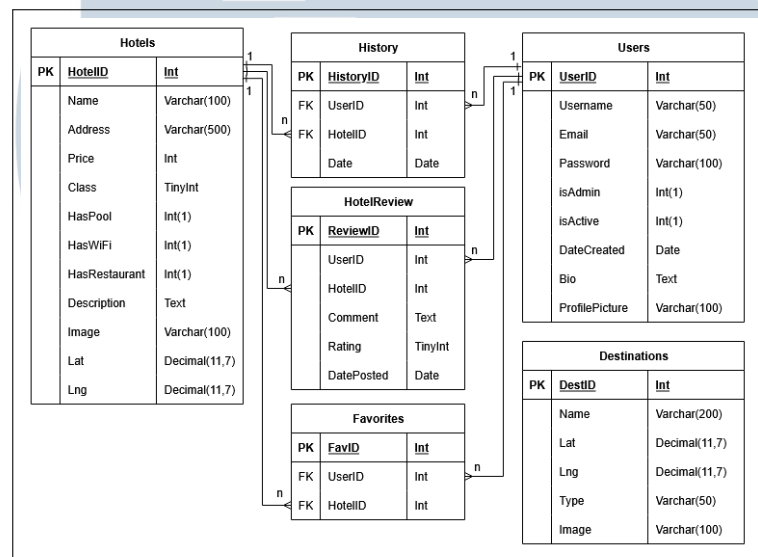


Gambar 3.14. Flowchart proses WPM untuk mencari alternatif terbaik

Flowchart pada Gambar 3.14 menggambarkan proses WPM setelah user menginput preferensi kriteria. Sistem akan menghitung berdasarkan data hotel, sehingga informasi hotel diambil dari *database* terlebih dahulu. Karena salah satu kriteria adalah jarak ke destinasi tertentu, maka data destinasi yang dipilih juga diambil, dan jarak antara setiap hotel dan destinasi tersebut dihitung. Dengan nilai setiap kriteria dan bobot yang didapatkan dari proses AHP, tahapan perhitungan WPM dimulai dengan normalisasi tabel data berdasarkan apakah kriteria merupakan *benefit* atau *cost* sesuai input user. Kemudian dilakukan perhitungan WPM yaitu pemangkatan dengan bobot dan perkalian nilai-nilai di alternatif yang sama, dan alternatif dengan nilai akhir tertinggi disimpan sebagai alternatif terbaik untuk direkomendasikan ke user.

3.4.3 Diagram Relasi Basis Data

Diagram Relasi Basis Data atau *Entity Relational Diagram* (ERD) merupakan diagram yang menggambarkan tabel-tabel dalam suatu basis data serta hubungan antar tabel. Setiap tabel memiliki kolom masing-masing dengan satu kolom menjadi *primary key*. Diagram skema untuk *website* sistem rekomendasi hotel adalah sebagai berikut:



Gambar 3.15. Diagram relasi basis data

Berdasarkan Gambar 3.15, ada enam entitas utama yang digunakan di *website* ini, yaitu user, hotel, destinasi, *history*, ulasan hotel, dan favorit. User terdiri dari email dan *password* untuk keperluan autentikasi, username, bio, dan foto profil yang bisa diubah di menu ubah profil, serta status apakah user tersebut merupakan admin dan user aktif (tidak diblokir). Hotel terdiri dari nama, alamat, harga standar (2 hari 1 malam untuk 2 orang dewasa), kelas, adanya kolam renang, restoran, dan Wi-Fi, deskripsi, gambar, dan lokasi lintang dan bujur untuk keperluan menghitung jarak ke destinasi. Data destinasi terdiri dari nama, tipe destinasi, gambar, dan lokasi lintang dan bujur. Data *history* terdiri dari user yang telah mendapatkan rekomendasi, hotel yang direkomendasikan oleh sistem, dan tanggal kapan user mendapatkan rekomendasi tersebut. Data ulasan terdiri dari ID User yang mengirim ulasan, ID hotel yang diulas, rating, komentar, dan tanggal dikirimnya ulasan. Data favorit terdiri dari ID user dan ID hotel yang dijadikan favorit.

Relasi antara tabel Users dan Hotels dengan History adalah *one-to-many*, dimana setiap data *history* memiliki satu hotel dan user, dan satu hotel dan user

dapat berada di beberapa data *history*. Relasi antara tabel Users dan Hotels dengan Reviews adalah *one-to-many*, dimana setiap data ulasan dikaitkan ke satu hotel dan dibuat oleh satu user, dan satu user bisa membuat banyak ulasan dan satu hotel bisa memiliki banyak ulasan. Relasi antara tabel Users dan Hotels dengan Favorites adalah *one-to-many*, dimana satu elemen pada tabel Favorites berhubungan dengan satu hotel dan satu user, sedangkan satu user dapat mengatur banyak hotel sebagai favorit, dan satu hotel dapat memiliki banyak user yang menjadikannya sebagai favorit. Tabel Favorites bukan entitas sendiri dan menjadi penghubung antara tabel Users dan Hotels, sehingga relasi antara tabel Users dan Hotels dalam konteks ini adalah *many-to-many*.

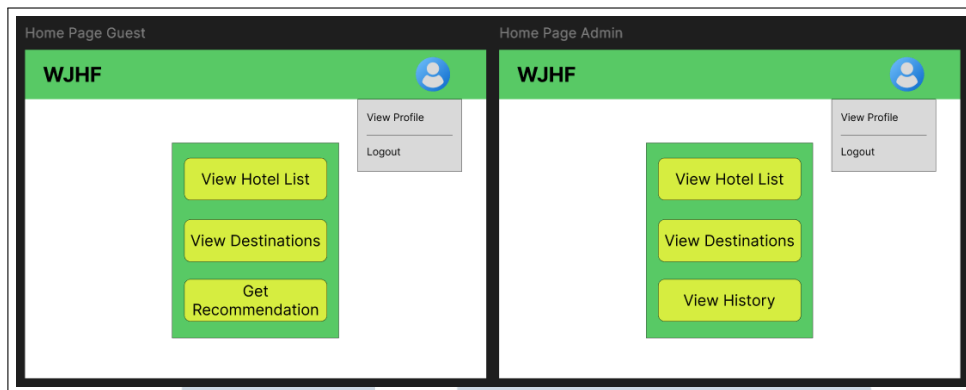
3.4.4 Rancangan Tampilan Website

Tahap perancangan tampilan *website* berfungsi untuk menentukan kerangka dan desain UI (*User Interface*) pada *website*. Rancangan ini dibuat menggunakan *mockup*, yaitu suatu konsep tampilan untuk sebuah sistem informasi, yang dirancang menggunakan *website* Figma.



Gambar 3.16. *Mockup* untuk tampilan halaman Login dan Registrasi

Gambar 3.16 merupakan *mockup* untuk tampilan halaman autentikasi, yaitu login di sebelah kiri dan registrasi di sebelah kanan. Ada beberapa *text box* yang dapat diisi oleh user untuk memasukkan data autentikasi, yaitu email dan *password* untuk proses login, dan ditambah *username* untuk registrasi. Kedua halaman ini juga memiliki *link* ke halaman autentikasi yang lainnya.



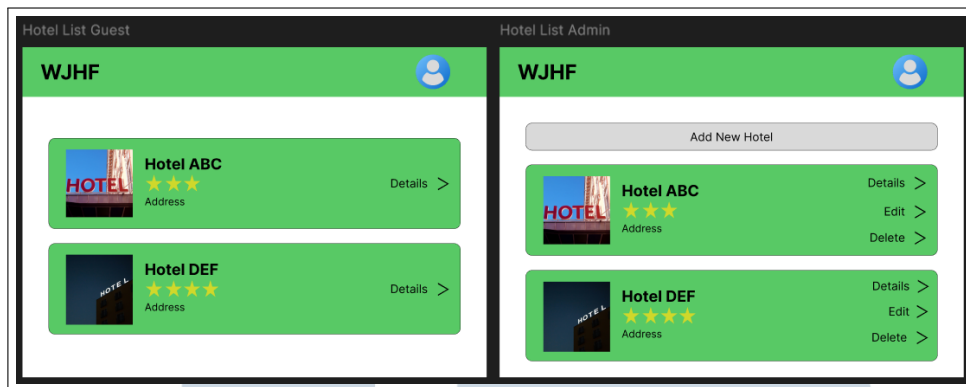
Gambar 3.17. *Mockup* untuk tampilan halaman menu utama

Gambar 3.17 merupakan *mockup* untuk tampilan menu utama. Di sebelah kiri adalah tampilan untuk user biasa, dan di sebelah kanan adalah tampilan untuk admin. Perbedaan utama menu untuk kedua peran adalah menu user memiliki opsi untuk mendapatkan rekomendasi, sedangkan menu admin memiliki opsi untuk melihat seluruh *history* rekomendasi. Ketika mengklik foto profil di *toolbar header*, ada juga opsi untuk melihat profil dan logout.



Gambar 3.18. *Mockup* untuk tampilan halaman profil

Gambar 3.18 merupakan *mockup* untuk tampilan profil, dimana user dapat melihat foto profil, *username*, email, tanggal akun dibuat, dan juga bio user yang dapat diedit menggunakan tombol Edit Profile. User juga dapat melihat ulasan yang telah dia berikan, hotel yang disukai, dan rekomendasi yang telah didapatkan user itu sendiri.



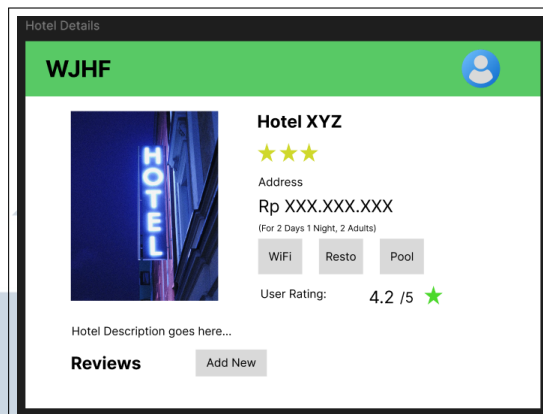
Gambar 3.19. *Mockup* untuk tampilan halaman daftar hotel

Gambar 3.19 merupakan *mockup* untuk daftar hotel. Seperti menu utama, tampilan daftar hotel memiliki perbedaan antara user (di sebelah kiri) dan admin (di sebelah kanan). Di menu ini, daftar hotel secara singkat diperlihatkan, dengan menampilkan gambar, nama hotel, kelas, dan alamat hotel. Admin dapat menambahkan hotel baru serta mengedit data hotel maupun menghapus hotel.

 The image shows a mockup of the 'Add/Edit Hotel' form. It has a green header with the 'WJHF' logo and a user profile icon. The form title is 'Add New Hotel'. It contains several input fields: 'Name:', 'Class:' (with a dropdown arrow), 'Price:', 'Address:', 'WiFi:' (with a dropdown arrow), 'Resto:' (with a dropdown arrow), 'Pool:' (with a dropdown arrow), 'Lat:', 'Lng:', 'Image:', and a 'Description:' text area. At the bottom right is a 'SUBMIT' button.

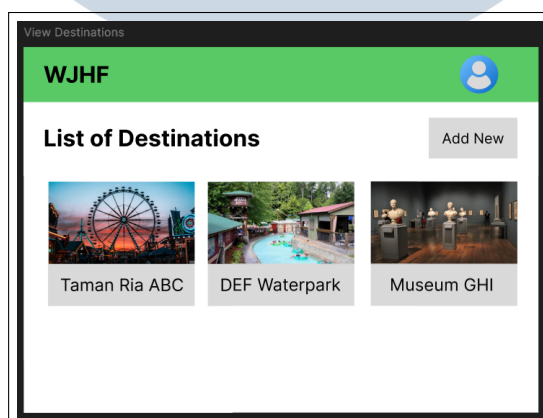
Gambar 3.20. *Mockup* untuk tampilan halaman hotel baru

Gambar 3.20 merupakan *mockup* untuk menu Tambah Hotel yang hanya dapat diakses oleh admin dan memiliki sebuah form dengan berbagai macam data hotel yang harus diisi oleh admin. Data-data tersebut yaitu nama hotel, kelas (jumlah bintang), harga, alamat, ketersediaan Wi-Fi, restoran, dan kolam renang, koordinat lintang dan bujur untuk proses perhitungan jarak ke destinasi, gambar dan deskripsi hotel. Menu edit hotel memiliki tampilan yang mirip.



Gambar 3.21. *Mockup* untuk tampilan halaman detail hotel

Gambar 3.21 merupakan *mockup* untuk menu detail hotel yang memiliki semua detail hotel secara lengkap, kecuali data lintang dan bujur. Selain itu, user (termasuk admin) juga bisa memberikan ulasan pada hotel, dimana daftar ulasan dapat dilihat di bawah deskripsi, serta rating rata-rata yang diberikan oleh user sistem rekomendasi ditampilkan di bawah data hotel.



Gambar 3.22. *Mockup* untuk tampilan halaman daftar destinasi

Gambar 3.22 merupakan *mockup* untuk halaman daftar destinasi. Tampilan daftar destinasi berbeda dengan daftar hotel, dimana destinasi akan ditampilkan dalam format *card*, dengan gambar di atas dan nama destinasi di bawah. Seperti untuk hotel, admin dapat menambahkan destinasi baru.

Recommendation (Criteria Priorities)

WJHF

Step 1: Determine Criteria Priorities

	Class	Price	Has WiFi	Has Restaurant	Has Pool	Distance to Dest.
Class	1	1	1	1	1	1
Price	1	1	1	1	1	1
Has WiFi	1	1	1	1	1	1
Has Restaurant	1	1	1	1	1	1
Has Pool	1	1	1	1	1	1
Distance to Dest.	1	1	1	1	1	1

SUBMIT

Gambar 3.23. *Mockup* untuk tampilan halaman tahap 1 dalam proses rekomendasi

Recommendation (Criteria Preferences)

WJHF

Step 2: Determine Criteria Preferences

Class: V Price Range: V

Desired Destination: V

WiFi: V Resto: V Pool: V

SUBMIT

Gambar 3.24. *Mockup* untuk tampilan halaman tahap 2 dalam proses rekomendasi

Recommendation Result

WJHF

Results

Based on your choices, we recommend you:

HOTEL

Hotel ABC

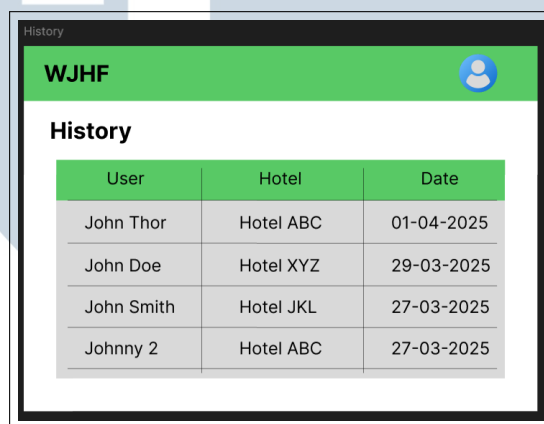
Address

Gambar 3.25. *Mockup* untuk tampilan halaman hasil rekomendasi

Ada tiga menu dalam proses rekomendasi. Tampilan pada Gambar 3.23 merupakan tampilan Tahapan 1, yaitu mencari bobot kriteria. Di sini, user

menginput prioritas kriteria dalam format matriks berpasangan ganda, dimana user menentukan angka untuk skala prioritas kriteria satu dengan yang lainnya. Ada juga tombol untuk memperlihatkan cara penggunaan dan detail-detail tentang mengisi matriks tersebut.

Kemudian, Gambar 3.24 berisi Tahapan 2, yaitu menentukan preferensi kriteria. Di sini, user menginput preferensi kriteria serta destinasi terdekat dari hotel yang diinginkan. Semua preferensi selain destinasi bersifat *benefit/cost* sebagai acuan dalam normalisasi kolom pada proses WPM. Ketika perhitungan selesai, maka sistem mengembalikan gambar dan nama hotel terbaik seperti pada Gambar 3.25.



User	Hotel	Date
John Thor	Hotel ABC	01-04-2025
John Doe	Hotel XYZ	29-03-2025
John Smith	Hotel JKL	27-03-2025
Johnny 2	Hotel ABC	27-03-2025

Gambar 3.26. *Mockup* untuk tampilan halaman *history*

Terakhir adalah *mockup* untuk tampilan semua *history* yang hanya dapat diakses oleh admin, seperti pada Gambar 3.26. Tampilan ini berbentuk tabel dengan data yaitu nama user yang melakukan rekomendasi, hotel yang direkomendasikan, dan tanggal proses rekomendasi tersebut dilakukan.

3.5 Contoh perhitungan rekomendasi hotel

Berikut ini merupakan contoh dari proses perhitungan untuk mendapatkan rekomendasi hotel. Proses ini dimulai dahulu dengan input user pada matriks berpasangan ganda untuk membandingkan prioritas kriteria. Tabel 3.2 merupakan contoh dari input user yang akan dikirim ke sistem untuk mencari bobot kriteria.

Tabel 3.2. Contoh input matriks berpasangan ganda untuk perbandingan kriteria

	C1	C2	C3	C4	C5	C6
C1	1	0.333	2	1	0.5	2
C2	3	1	2	2	1	4
C3	0.5	0.5	1	2	0.5	2
C4	1	0.5	0.5	1	0.5	0.333
C5	2	1	2	2	1	3
C6	0.5	0.25	0.5	3	0.33	1
Jumlah	8	3.583	8	11	3.833	12.333

Dimana C_x merupakan representasi dari masing-masing kriteria dalam penelitian. C1 merupakan kelas hotel, C2 merupakan harga, C3 merupakan adanya Wi-Fi, C4 merupakan adanya restoran, C5 merupakan adanya kolam renang, dan C6 merupakan jarak dari destinasi yang nantinya akan dipilih di Tahap 2.

Setelah input user dikirim ke sistem, sistem akan melakukan tahap perhitungan bobot kriteria menggunakan metode AHP. Pertama-tama, matriks berpasangan ganda akan dinormalisasi dengan membagi nilai setiap elemen pada matriks dengan jumlah kolom masing-masing, dengan Rumus 3.1a, menghasilkan nilai pada Tabel 3.3. Kemudian, vektor eigen setiap kriteria dicari dengan menjumlahkan nilai setiap baris dan mencari rata-ratanya, dengan Rumus 3.1b. Nilai rata-rata ini akan menjadi bobot sementara untuk kriteria pada sistem rekomendasi.

$$dn_{ij} = \frac{d_{ij}}{\sum_{i=1}^n d_{ij}} \quad \text{Weight} = \frac{\sum_{j=1}^m dn_{ij}}{m} \quad (3.1)$$

Tabel 3.3. Matriks berpasangan ganda yang telah dinormalisasi

	C1	C2	C3	C4	C5	C6	Bobot
C1	0.125	0.093	0.25	0.091	0.13	0.162	0.142
C2	0.375	0.279	0.25	0.182	0.261	0.324	0.279
C3	0.063	0.14	0.125	0.182	0.13	0.162	0.134
C4	0.125	0.14	0.063	0.091	0.13	0.027	0.096
C5	0.25	0.279	0.25	0.182	0.261	0.243	0.244
C6	0.063	0.07	0.063	0.273	0.087	0.081	0.106

Walaupun secara teknis bobot kriteria telah didapatkan, namun sistem harus menguji konsistensi matriks berpasangan agar dapat menentukan apakah bobot yang dihasilkan pantas digunakan dalam tahapan kedua. Tahap pertama

dari uji konsistensi adalah mencari nilai Ax setiap kriteria yang didapatkan dengan mengalikan nilai setiap elemen pada matriks berpasangan ganda, dan menjumlahkan semua nilai per baris, seperti pada Rumus 3.2. Hasil perhitungan Ax ada pada Tabel 3.4.

$$Ax = \sum_{j=1}^m d_{ij} \times w_j \quad (3.2)$$

Tabel 3.4. Matriks berpasangan ganda berbobot yang telah dinormalisasi serta nilai Ax masing-masing kriteria

	C1	C2	C3	C4	C5	C6	Ax
Bobot	0.142	0.279	0.134	0.096	0.244	0.106	
C1	0.142	0.093	0.267	0.096	0.122	0.212	0.932
C2	0.426	0.279	0.267	0.192	0.244	0.424	1.831
C3	0.071	0.139	0.133	0.192	0.122	0.212	0.87
C4	0.142	0.139	0.067	0.096	0.122	0.035	0.601
C5	0.284	0.279	0.267	0.192	0.244	0.318	1.583
C6	0.071	0.07	0.067	0.288	0.081	0.106	0.682

Kemudian, nilai lambda dicari dengan membagi setiap nilai pada Ax dengan nilai bobot yang bersangkutan dengan kriteria/baris tersebut, dan mencari rata-rata dari hasil perhitungan tersebut, seperti pada Rumus 3.3. Berdasarkan nilai Ax di atas, nilai lambda yang didapatkan adalah **6.474**. Lalu, nilai *consistency index* dihitung menggunakan rumus seperti ada pada Rumus 2.1. Didapatkan nilai CI adalah **0.095**. Terakhir, nilai *consistency ratio* dihitung menggunakan Rumus 2.2, dimana *random index* yang digunakan bernilai 1.24 karena ada 6 kriteria.

$$\lambda = \frac{\sum \frac{Ax_i}{w_j}}{n} \quad (3.3)$$

Tabel 3.5. Hasil perhitungan lambda, CI, dan CR dalam uji konsistensi

Lambda	Consistency Index	Consistency Ratio
6.473925	0.094785	0.07644

Nilai CR yang didapatkan adalah **0.076**. Karena nilai ini lebih kecil dari *threshold* penerimaan konsistensi matriks yaitu 0.1, maka matriks berpasangan ganda ini dinilai konsisten, dan bobot yang telah didapatkan dapat digunakan

dalam tahapan selanjutnya, yaitu mencari alternatif terbaik. Bobot akhir setelah perhitungan AHP ada pada Tabel 3.6.

Tabel 3.6. Bobot akhir kriteria setelah perhitungan metode AHP

Nama Kriteria	Bobot
Kelas	0.142
Harga	0.279
Ada Wi-Fi	0.134
Ada Restoran	0.096
Ada Kolam Renang	0.244
Jarak ke Destinasi yang Diinginkan	0.106

Dalam perhitungan mencari alternatif terbaik, digunakan metode WPM, dengan data yang digunakan dalam perhitungan berupa data numerik hotel, yaitu harga, kelas, jarak, dan ketersediaan fasilitas. Dalam simulasi perhitungan, diambil 5 data hotel secara acak dari *database*, dengan data seperti pada Tabel 3.7.

Tabel 3.7. Contoh data hotel untuk perhitungan alternatif terbaik

Nama hotel	Kelas	Harga	Ada Wi-Fi	Ada Resto	Ada Kolam Renang	Posisi Lintang	Posisi Bujur
Hotel Couleur	3	300000	1	1	0	-6.1677956	106.7252777
Harris Suites Puri Mansion	4	780000	1	1	1	-6.1796090	106.7217874
RedDoorz near Seasons City Mall	2	203000	1	0	0	-6.1550534	106.7876995
D'Paragon Kebon Jeruk Jakarta	2	265000	1	0	0	-6.1916584	106.7594424
Mega Anggrek Hotel	3	300000	1	1	1	6.1850629	106.7872891

Setelah bobot kriteria dari perhitungan AHP didapatkan, user memilih preferensi kriteria. Preferensi untuk setiap kriteria dapat bersifat *benefit* atau

cost, dimana pilihan user menentukan apakah sistem akan memprioritaskan nilai lebih tinggi atau rendah. Selain itu, user juga dapat memilih destinasi terdekat dengan hotel yang diinginkan, dimana jarak dari hotel ke destinasi dinilai sebagai atribut *cost*. Tabel 3.8 merupakan sampel input user pada tahap kedua perhitungan rekomendasi.

Tabel 3.8. Contoh pilihan preferensi user untuk mencari alternatif terbaik

Nama Kriteria	Pilihan user	Jenis atribut
Kelas	Utamakan hotel berkelas tinggi	<i>Benefit</i>
Harga	Utamakan hotel murah	<i>Cost</i>
Ada Wi-Fi	Harus ada	<i>Benefit</i>
Ada Restoran	Harus ada	<i>Benefit</i>
Ada Kolam Renang	Tidak perlu ada	<i>Cost</i>
Jarak ke Destinasi yang Diinginkan	Bandara Soekarno-Hatta	<i>Cost</i>

Kemudian, sistem akan membuat tabel untuk pelaksanaan perhitungan. Atribut kelas dan harga tetap sama, atribut fasilitas ditambahkan 1 agar menghindari pembagian dan perkalian dengan nol, dan atribut jarak ditentukan dengan menghitung jarak antara hotel dengan destinasi pilihan menggunakan lokasi lintang dan bujur. Jarak dihitung menggunakan rumus *Haversine distance*, yang ditetapkan sebagai berikut:

$$D = 2 \times \sin^{-1} \left(\sqrt{\sin^2\left(\frac{lat2 - lat1}{2}\right) + \sin^2\left(\frac{lng2 - lng1}{2}\right) \times \cos(lat1) \times \cos(lat2)} \right) \times \mu_E \quad (3.4)$$

Dimana *lat1* dan *lng1* merupakan posisi lintang dan bujur dari hotel, *lat2* dan *lng2* merupakan posisi lintang dan bujur dari destinasi, dan μ_e merupakan jari-jari dari bumi, yaitu 6371000 m. Dalam perhitungan ini, destinasi yang dipilih adalah Bandara Soekarno Hatta yang terletak pada (-6.1265511, 106.6468116). Jarak akan disimpan dalam satuan meter. Data hotel untuk perhitungan WPM, bersama dengan jarak ke Bandara Soekarno-Hatta, dan bobot masing-masing kriteria, terdapat pada Tabel 3.9. Nilai terbaik setiap kolom ditentukan dengan tipe kolom: jika tipe kolom *benefit*, maka cari nilai terbesar, dan jika tipe *cost*, maka cari nilai terkecil.

Tabel 3.9. Data hotel setelah perhitungan jarak ke bandara

Nama hotel	Kelas (B)	Harga (C)	Ada Wi-Fi (B)	Ada Resto (B)	Ada Kolam Renang (C)	Jarak ke bandara (C)
Bobot	0.142	0.279	0.134	0.096	0.244	0.106
Hotel Couleur	3	300000	2	2	1	9813
Harris Suites Puri Mansion	4	780000	2	2	2	10174
RedDoorz near Seasons City Mall	2	203000	2	1	1	15895
D'Paragon Kebon Jeruk Jakarta	2	265000	2	1	1	14401
Mega Anggrek Hotel	3	300000	2	2	2	16838
Nilai terbaik	4	265000	2	2	1	9813

Tahap pertama setelah mendapatkan tabel tersebut adalah normalisasi tabel. Untuk kriteria yang bersifat *benefit*, normalisasi dilakukan dengan membagi elemen tabel dengan nilai maksimum pada kolom kriteria. Untuk kriteria yang bersifat *cost*, normalisasi dilakukan dengan membagi nilai minimum pada kolom kriteria dengan elemen tabel. Data hotel setelah normalisasi ada pada Tabel 3.10.

Tabel 3.10. Data hotel setelah normalisasi kolom

Nama hotel	Kelas	Harga	Ada Wi-Fi	Ada Resto	Ada Kolam Renang	Jarak ke bandara
Bobot	0.142	0.279	0.134	0.096	0.244	0.106
Hotel Couleur	0.75	0.883	1	1	1	1
Harris Suites Puri Mansion	1	0.353	1	1	0.5	0.965
RedDoorz near Seasons City Mall	0.5	0.589	1	0.5	1	0.617
D'Paragon Kebon Jeruk Jakarta	0.5	1	1	0.5	1	0.681
Mega Anggrek Hotel	0.75	0.883	1	1	0.5	0.583

Tahap selanjutnya adalah dengan mengangkatkan setiap elemen pada tabel dengan bobot setiap kolom, dengan hasil seperti pada Tabel 3.11.

Tabel 3.11. Data hotel setelah dipangkatkan dengan bobot kriteria

Nama hotel	Kelas	Harga	Ada Wi-Fi	Ada Resto	Ada Kolam Renang	Jarak ke bandara
Bobot	0.142	0.279	0.134	0.096	0.244	0.106
Hotel Couleur	0.96	0.966	1	1	1	1
Harris Suites Puri Mansion	1	0.748	1	1	0.844	0.996
RedDoorz near Seasons City Mall	0.906	0.863	1	0.936	1	0.95
D'Paragon Kebon Jeruk Jakarta	0.906	1	1	0.936	1	0.96
Mega Anggrek Hotel	0.96	0.966	1	1	0.844	0.944

Tahap akhir adalah dengan mengalikan semua nilai atribut per baris, untuk mendapatkan nilai rekomendasi untuk setiap hotel. Nilai rekomendasi yang paling

tinggi akan dipilih sebagai hotel terbaik. Data nilai akhir hotel ada pada Tabel 3.12.

Tabel 3.12. Data nilai rekomendasi akhir untuk setiap hotel

Peringkat	Nama hotel	Nilai perkalian semua atribut
1	Hotel Couleur	0.927315
2	D'Paragon Kebon Jeruk Jakarta	0.814135
3	Mega Anggrek Hotel	0.73947
4	RedDoorz near Seasons City Mall	0.69501
5	Harris Suites Puri Mansion	0.629258

Berdasarkan Tabel 3.12, nilai tertinggi dimiliki oleh Hotel Couleur. Maka, sistem akan mengembalikan Hotel Couleur sebagai rekomendasi hotel kepada user yang sesuai dengan prioritas dan pilihan kriteria yang diinginkan.

