

LAPORAN UAS

**Perancangan Sistem Informasi TRAVELOOP: Sistem Perjalanan Adaptif Berbasis
Kecerdasan Buatan untuk Perencanaan Itinerary Otomatis dan Real-Time**

Disusun untuk Memenuhi Matakuliah Software Modeling
Dibimbing oleh Ibu Solehatin M.Kom



Oleh:

Rhegysa Alvyanthi Juniarta - 1123102098

Imas Nabellia Venda - 1123102116

**PROGRAM STUDI S1 TEKNIK INFORMATIKA
SEKOLAH TINGGI ILMU KOMPUTER PGRI BANYUWANGI
2025**

I. Analisa Permasalahan

1. Alasan Pemilihan Judul

Pemilihan judul ini didasarkan pada kebutuhan akan sistem yang mampu memberikan pengalaman perjalanan yang lebih efisien, personal, dan adaptif terhadap perubahan kondisi di lapangan. Dalam era digital, masyarakat semakin bergantung pada teknologi untuk mempermudah aktivitas sehari-hari, termasuk dalam hal perjalanan. Namun, kebanyakan aplikasi perjalanan hanya menyediakan fitur statis seperti pemesanan tiket dan akomodasi tanpa adanya penyesuaian otomatis terhadap perubahan yang terjadi selama perjalanan. Oleh karena itu, pengangkatan judul ini bertujuan untuk merancang sistem yang tidak hanya memberikan rekomendasi perjalanan, tetapi juga dapat menyesuaikan jadwal secara real-time berdasarkan berbagai faktor seperti cuaca, keterlambatan transportasi, dan preferensi pengguna.

2. Input

- a. Data pengguna seperti nama, email, histori perjalanan.
- b. Preferensi perjalanan seperti jenis destinasi favorit, gaya perjalanan, estimasi anggaran.
- c. Lokasi dan waktu perjalanan seperti titik keberangkatan, destinasi tujuan, tanggal perjalanan, durasi perjalanan.
- d. Data cuaca seperti informasi cuaca real-time pada setiap destinasi.
- e. Data transportasi seperti jadwal transportasi, ketersediaan transportasi, dan status transportasi.
- f. Data peta dan lokasi seperti koordinat lokasi, jarak antar destinasi, waktu tempuh.
- g. Harga dan biaya seperti harga tiket wisata, penginapan, konsumsi, estimasi transportasi lokal.

3. Proses

- a. Pengguna melakukan login ke dalam sistem.
- b. Sistem mengambil data pengguna dari basis data.
- c. Jika pengguna baru, maka sistem menampilkan formulir input mengisi preferensi perjalanan seperti jenis destinasi, anggaran, gaya perjalanan.

- d. Jika bukan pengguna baru, maka sistem mengambil histori preferensi dan perilaku perjalanan dari basis data.
- e. Sistem mengklasifikasikan preferensi pengguna ke dalam kategori destinasi, seperti wisata alam, budaya, atau kuliner.
- f. Sistem menentukan prioritas destinasi berdasarkan riwayat perjalanan pengguna, batasan anggaran pengguna, dan gaya perjalanan seperti hemat, santai, atau eksploratif.
- g. Sistem mengakses API eksternal untuk memperoleh data cuaca terkini, informasi transportasi, dan peta lokasi.
- h. Data yang diperoleh disinkronkan secara real-time.
- i. Jika kondisi cuaca pada destinasi utama baik, maka destinasi tersebut dimasukkan ke dalam rencana perjalanan.
- j. Jika cuaca buruk, maka sistem otomatis mengganti dengan destinasi alternatif indoor atau mengatur ulang waktu kunjungan.
- k. Jika transportasi tersedia dan sesuai waktu yang direncanakan, maka sistem melanjutkan proses.
- l. Jika tidak tersedia, maka sistem mencari transportasi alternatif atau mengubah urutan destinasi.
- m. Sistem menghitung total estimasi biaya perjalanan, termasuk akomodasi, transportasi, tiket masuk, dan konsumsi harian.
- n. Jika total estimasi biaya melebihi anggaran, maka sistem melakukan penyesuaian jumlah destinasi atau mengganti pilihan tempat yang lebih ekonomis.
- o. Jika sesuai, maka proses dilanjutkan tanpa perubahan.
- p. Sistem menyusun urutan kunjungan berdasarkan efisiensi jarak lokasi, waktu tempuh, kondisi cuaca, dan jadwal operasional masing-masing destinasi.
- q. Sistem menyimpan hasil itinerary.
- r. Sistem menampilkan itinerary adaptif kepada pengguna beserta estimasi biaya dan waktu.
- s. Selama perjalanan berlangsung sistem secara kontinu memantau kondisi cuaca terkini, posisi pengguna melalui GPS dan status transportasi.
- t. Jika terjadi gangguan maka sistem akan mengirimkan notifikasi ke pengguna dan menawarkan penyesuaian itinerary secara otomatis.

- u. Jika tidak terjadi gangguan maka sistem melanjutkan itinerary sesuai rencana.

4. Output

- a. Itinerary perjalanan adaptif yaitu rencana perjalanan harian yang sudah disesuaikan dengan preferensi dan kondisi actual.
- b. Estimasi total biaya perjalanan yaitu estimasi biaya menyeluruh seperti akomodasi, tiket, makan, dan transportasi lokal.
- c. Rekomendasi destinasi yaitu destinasi yang direkomendasikan sistem berdasarkan preferensi pengguna dan ketersediaan lokasi.
- d. Notifikasi perubahan rencana yaitu pemberitahuan jika terjadi cuaca buruk, pembatalan transportasi, atau perubahan itinerary.
- e. Alternatif itinerary ketika ada gangguan yaitu opsi itinerary baru yang ditawarkan kepada pengguna saat kondisi tidak memungkinkan mengikuti rencana awal.
- f. Riwayat perjalanan yaitu data perjalanan yang sudah diselesaikan oleh pengguna untuk keperluan pembelajaran sistem.

5. User dalam Sistem

- a. Traveler yaitu pengguna utama yang merencanakan perjalanan melalui pengisian preferensi dan menerima itinerary otomatis dari sistem.
- b. Admin yang bertanggung jawab atas pengelolaan data destinasi wisata, akomodasi, dan transportasi. Selain itu admin juga bertanggung jawab atas validasi akun pengguna, pengaturan hak akses, serta pengawasan keamanan sistem. Admin mengelola integrasi API eksternal, menyusun konten tips perjalanan, dan mengatur notifikasi sistem seperti info cuaca dan perubahan itinerary.

6. Data yang Diperlukan

- a. Sistem
 - 1) Nama lengkap, email, nomor telepon, lokasi keberangkatan, preferensi gaya perjalanan, budget perjalanan.
 - 2) Nama destinasi, jenis wisata, Lokasi geografis, jam operasional destinasi, harga tiket masuk.

- 3) Nama penginapan, lokasi penginapan, kategori penginapan, harga per malam, ketersediaan kamar, rating pelayanan.
- 4) Jenis transportasi, jadwal keberangkatan dan kedatangan, harga tiket transportasi, ketersediaan kursi.
- 5) Prakiraan cuaca per hari dan per Lokasi destinasi yang dipilih traveler selama periode perjalanan yang sudah ditentukan.
- 6) Koordinat destinasi, waktu tempuh antar lokasi, rekomendasi jalur efisien antar destinasi.
- 7) Riwayat destinasi yang pernah dikunjungi traveler, jumlah hari liburan sebelumnya, frekuensi perjalanan, pola perjalanan traveler.
- 8) Rincian estimasi biaya perjalanan berdasarkan akumulasi komponen transportasi, akomodasi, konsumsi dan aktivitas.

b. Traveler

- 1) Nama lengkap
- 2) Alamat email
- 3) Nomor telepon
- 4) Lokasi keberangkatan
- 5) Tanggal keberangkatan dan kepulangan
- 6) Total budget yang dimiliki
- 7) Preferensi gaya perjalanan (hemat, santai, lengkap)
- 8) Pilihan tipe destinasi
- 9) Riwayat perjalanan dan revisi itinerary sebelumnya
- 10) Preferensi bahasa dan zona waktu

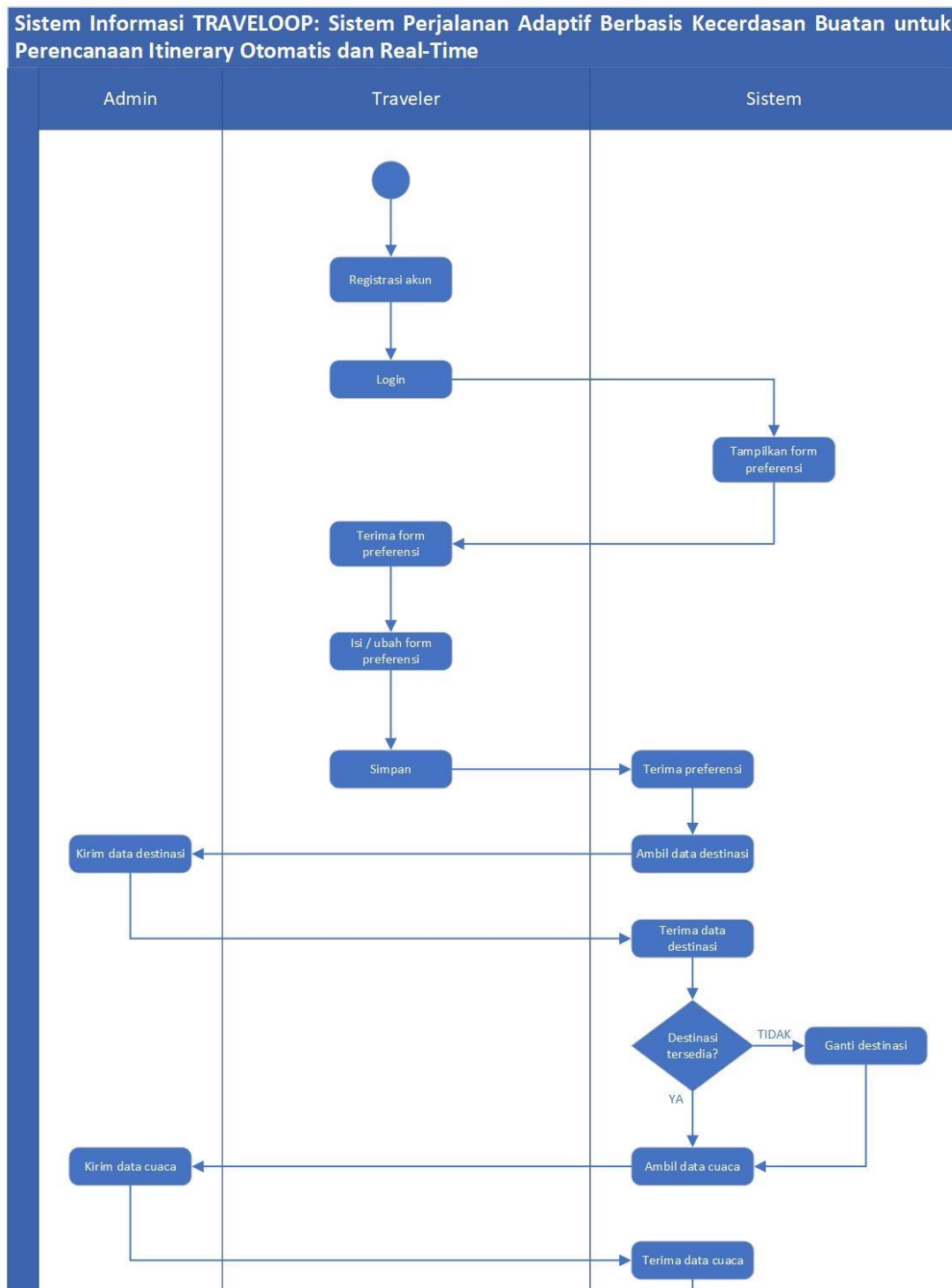
7. Yang Dilakukan Traveler dalam Sistem

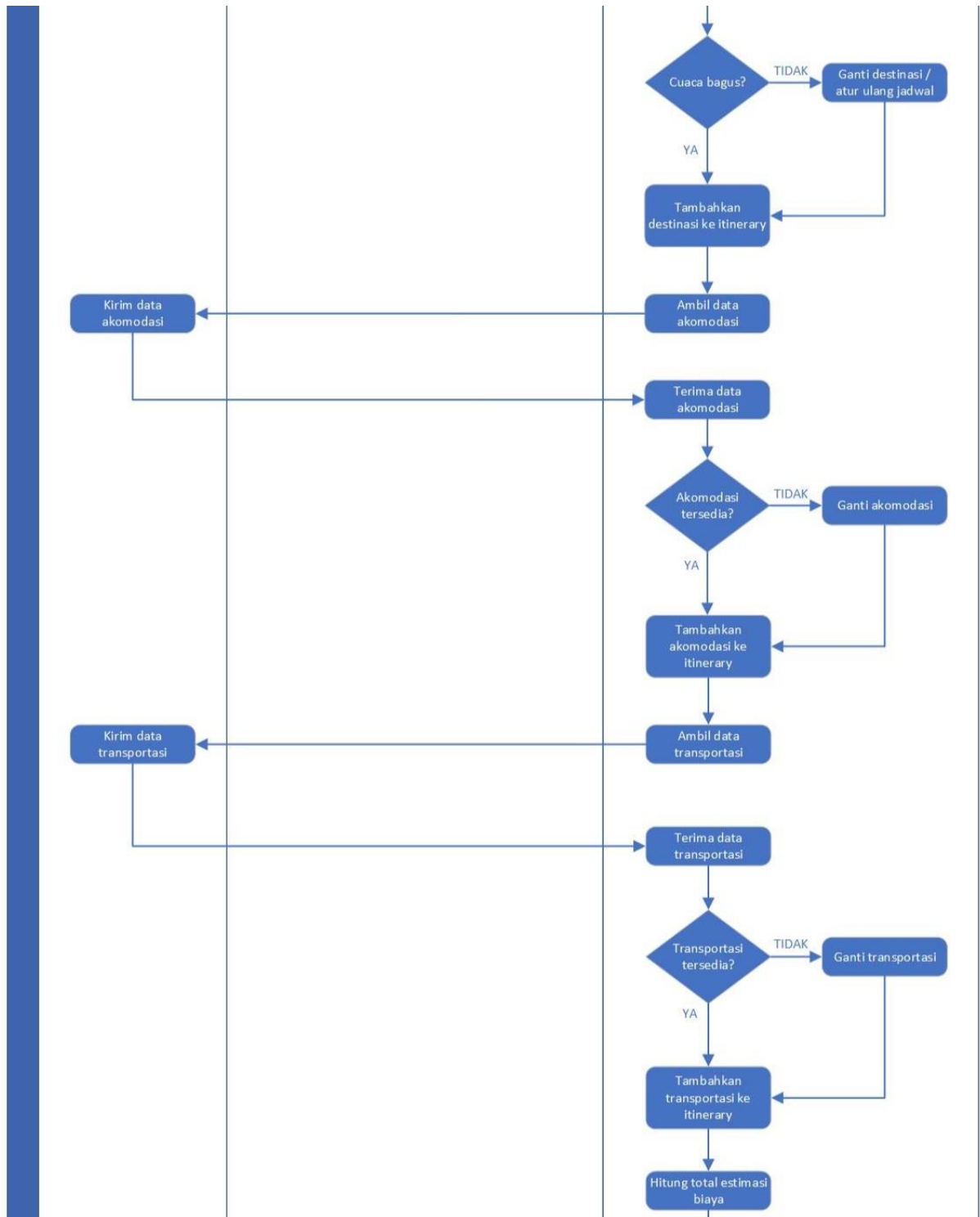
- a. Membuat dan melakukan login akun TRAVELOOP
- b. Mengisi form perjalanan
- c. Melihat rekomendasi itinerary otomatis yang disusun oleh sistem
- d. Mengedit dan menyesuaikan rekomendasi itinerary secara manual jika diperlukan

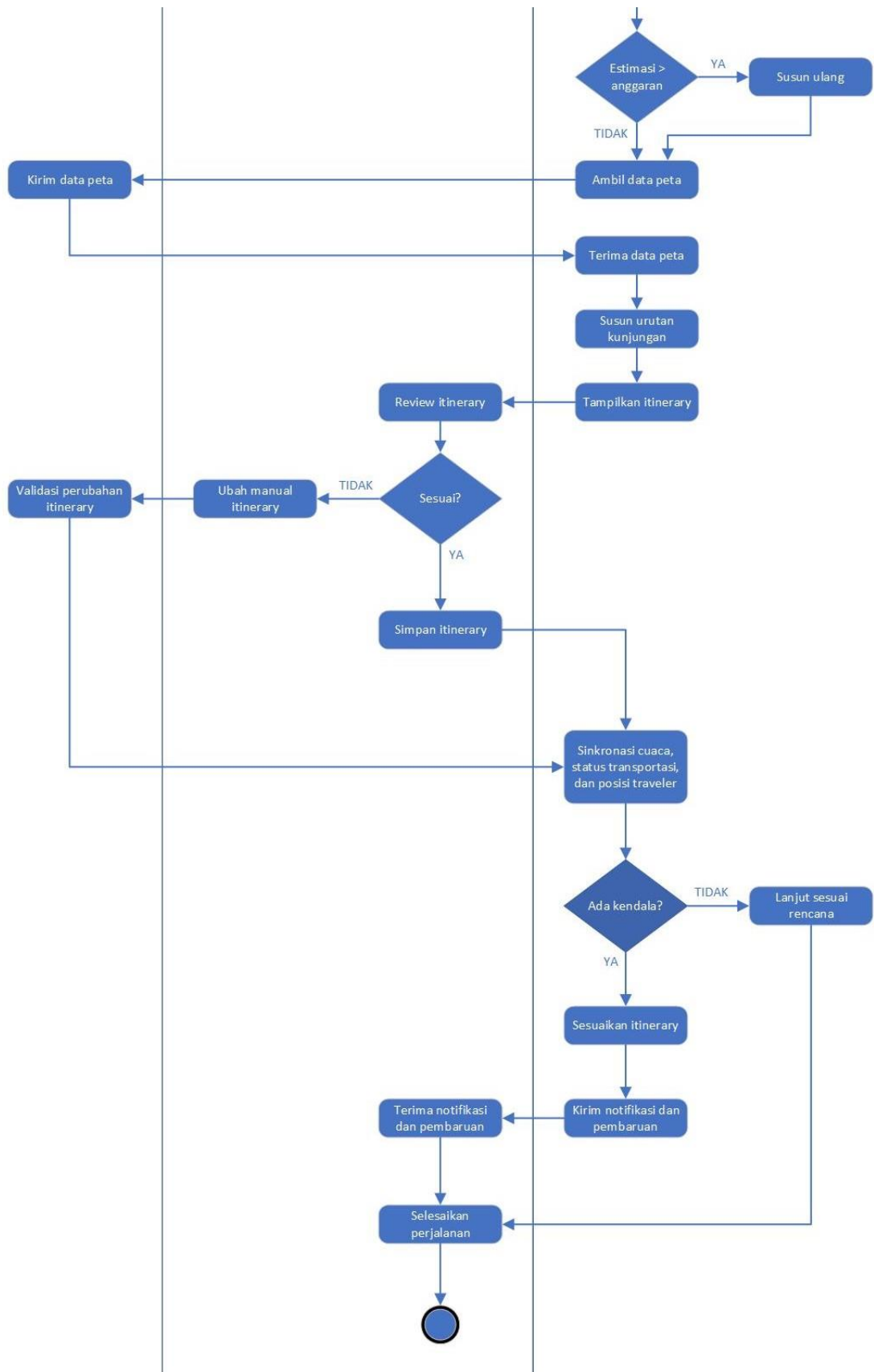
- e. Menyimpan itinerary yang disetujui ke akun pengguna dan mengunduhnya dalam format PDF atau melalui aplikasi
- f. Menerima notifikasi realtime mengenai cuaca ekstrem, perubahan jadwal transportasi, atau promo mendadak
- g. Melakukan penyesuaian realtime terhadap perjalanan berdasarkan kondisi actual selama trip

II. Bisnis Proses

Sistem Informasi TRAVELOOP adalah platform perencanaan perjalanan berbasis kecerdasan buatan yang bertujuan untuk menyusun itinerary (rencana perjalanan) secara otomatis dan real-time berdasarkan preferensi pengguna, kondisi cuaca, ketersediaan layanan, serta estimasi biaya. Bisnis proses dalam sistem ini terdiri dari beberapa tahapan yang terintegrasi dimulai dari interaksi awal pengguna hingga selesainya perjalanan.







Tahapan pertama dimulai ketika pengguna melakukan proses login ke dalam sistem. Apabila pengguna merupakan pengguna baru sistem akan menampilkan formulir untuk mengisi preferensi perjalanan seperti jenis destinasi, aktivitas yang disukai, anggaran, durasi perjalanan, dan lain sebagainya. Jika pengguna sudah pernah menggunakan sistem sebelumnya sistem akan secara otomatis mengambil dan menampilkan data preferensi yang tersimpan, dan memberikan opsi untuk mengubah jika diperlukan.

Setelah preferensi diperoleh, sistem akan mengakses API destinasi untuk mengambil data tempat-tempat yang sesuai dengan preferensi tersebut. Data yang diterima akan ditampilkan kepada pengguna dan pengguna dapat mengganti destinasi jika pilihan yang diberikan tidak sesuai. Kemudian, sistem akan mengakses API cuaca untuk mendapatkan informasi kondisi cuaca di destinasi yang dipilih. Jika cuaca dinilai tidak mendukung, sistem akan menyarankan pengguna untuk mengganti destinasi atau melakukan penjadwalan ulang.

Langkah berikutnya adalah menyusun itinerary dengan menambahkan komponen-komponen pendukung yang dimulai dari akomodasi. Sistem mengambil data akomodasi dari API yang relevan lalu menampilkan opsi kepada pengguna. Apabila akomodasi tidak tersedia, akan dipilih akomodasi lain hingga tersedia yang sesuai. Proses yang sama diterapkan untuk transportasi, data diambil dari API transportasi dan disesuaikan dengan rencana perjalanan.

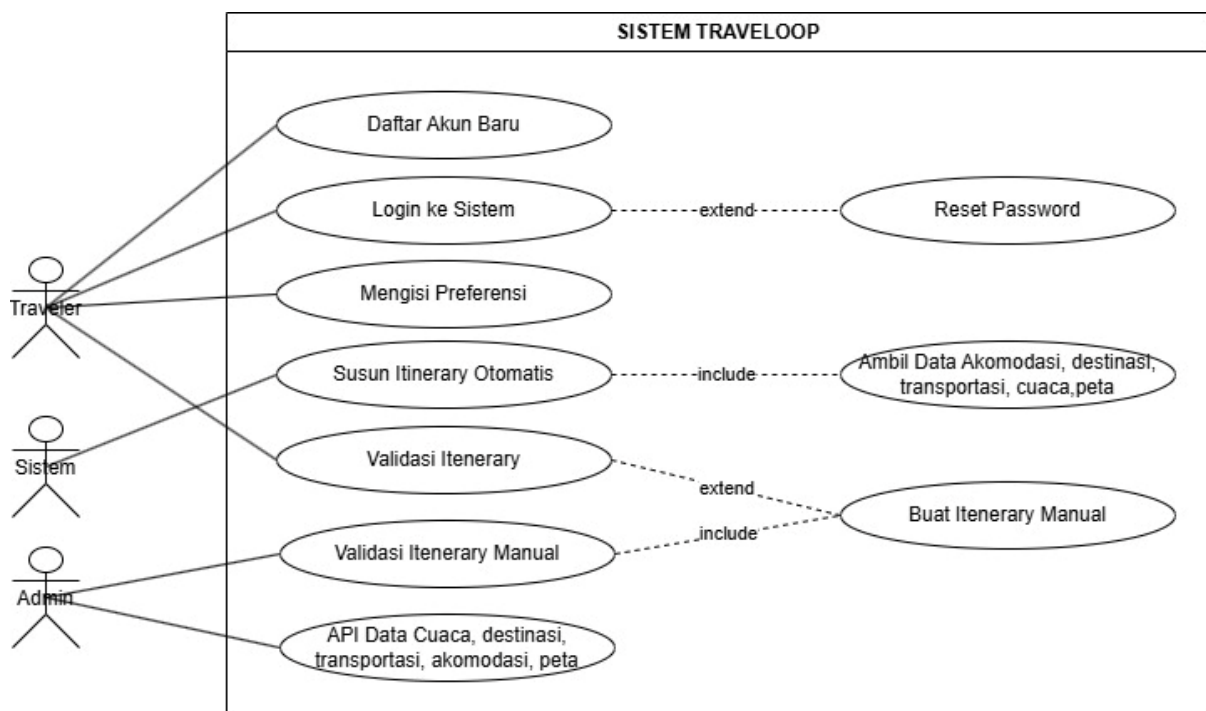
Setelah seluruh elemen perjalanan (destinasi, akomodasi, dan transportasi) tersedia dan ditambahkan ke dalam itinerary, sistem akan menghitung estimasi total biaya. Jika estimasi biaya melebihi anggaran yang ditentukan oleh pengguna maka sistem akan menyusun ulang itinerary secara otomatis untuk menyesuaikannya. Kemudian sistem menyusun urutan kunjungan dan mengambil data dari API peta untuk mendapatkan rute terbaik berdasarkan lokasi dan efisiensi perjalanan.

Setelah itinerary disusun lengkap sistem menampilkan hasilnya kepada pengguna untuk direview. Pengguna dapat melakukan perubahan secara manual dan sistem akan memvalidasi setiap perubahan tersebut. Setelah disetujui, itinerary akan disimpan dalam sistem.

Selama perjalanan berlangsung TRAVELOOP melakukan sinkronisasi data secara real-time, termasuk pembaruan cuaca, status transportasi, serta posisi pengguna. Jika terdapat kendala sistem akan menyesuaikan itinerary secara otomatis dan mengirimkan notifikasi pembaruan kepada pengguna. Proses ini berlangsung hingga perjalanan selesai yang kemudian ditandai sebagai tahapan akhir dari bisnis proses TRAVELOOP.

III. USE CASE

Use case diagram sistem TRAVELOOP menggambarkan interaksi antara aktor dengan sistem dalam merencanakan dan mengelola perjalanan wisata secara digital. Diagram ini dirancang untuk memetakan fitur-fitur utama yang tersedia dalam sistem, serta menunjukkan peran masing-masing aktor dalam menjalankan proses perencanaan itinerary secara otomatis maupun manual. Dalam sistem TRAVELOOP, terdapat tiga aktor utama yaitu Traveler (pengguna), Sistem, dan Admin.



Traveler adalah pengguna utama sistem yang memanfaatkan layanan untuk merencanakan perjalanan. Mereka dapat mendaftar akun baru, melakukan login, mengatur preferensi perjalanan, dan menggunakan fitur penyusunan itinerary otomatis berdasarkan data akomodasi, destinasi, transportasi, cuaca, dan peta. Selain itu, traveler juga dapat memvalidasi itinerary yang telah disusun, serta membuat itinerary manual apabila hasil otomatis tidak sesuai kebutuhan mereka.

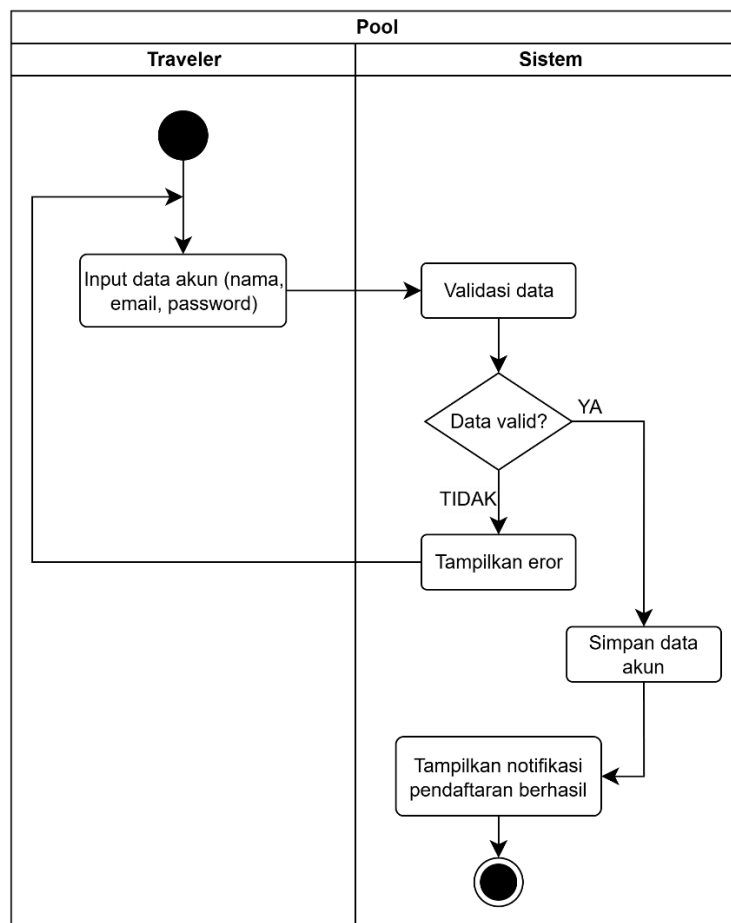
Sistem berperan sebagai entitas otomatis yang mengeksekusi proses internal, seperti mengambil data dari API eksternal (akomodasi, destinasi, transportasi, cuaca, dan peta), serta menyusun itinerary secara otomatis sesuai preferensi pengguna. Sistem juga mendukung validasi itinerary untuk membantu traveler dalam memastikan kesesuaian rencana perjalanan.

Admin bertanggung jawab atas pengelolaan dan integrasi data dari API eksternal. Admin juga melakukan validasi terhadap itinerary manual yang dibuat oleh pengguna, memastikan bahwa informasi yang digunakan dalam rencana perjalanan tersebut akurat dan relevan. Ketiga aktor ini saling berinteraksi untuk memberikan pengalaman perencanaan perjalanan yang terintegrasi dan efisien.

IV. Activity Diagram

Untuk menggambarkan bagaimana sistem Traveloop bekerja dalam membantu pengguna merencanakan perjalanan, activity diagram disusun untuk menjelaskan alur aktivitas di dalam aplikasi. Diagram-diagram ini mencakup proses penting mulai dari pendaftaran akun, login, pengisian preferensi, pembentukan itinerary otomatis, validasi real-time selama perjalanan, hingga pembuatan itinerary custom dan integrasi dengan API eksternal.

1. Registrasi



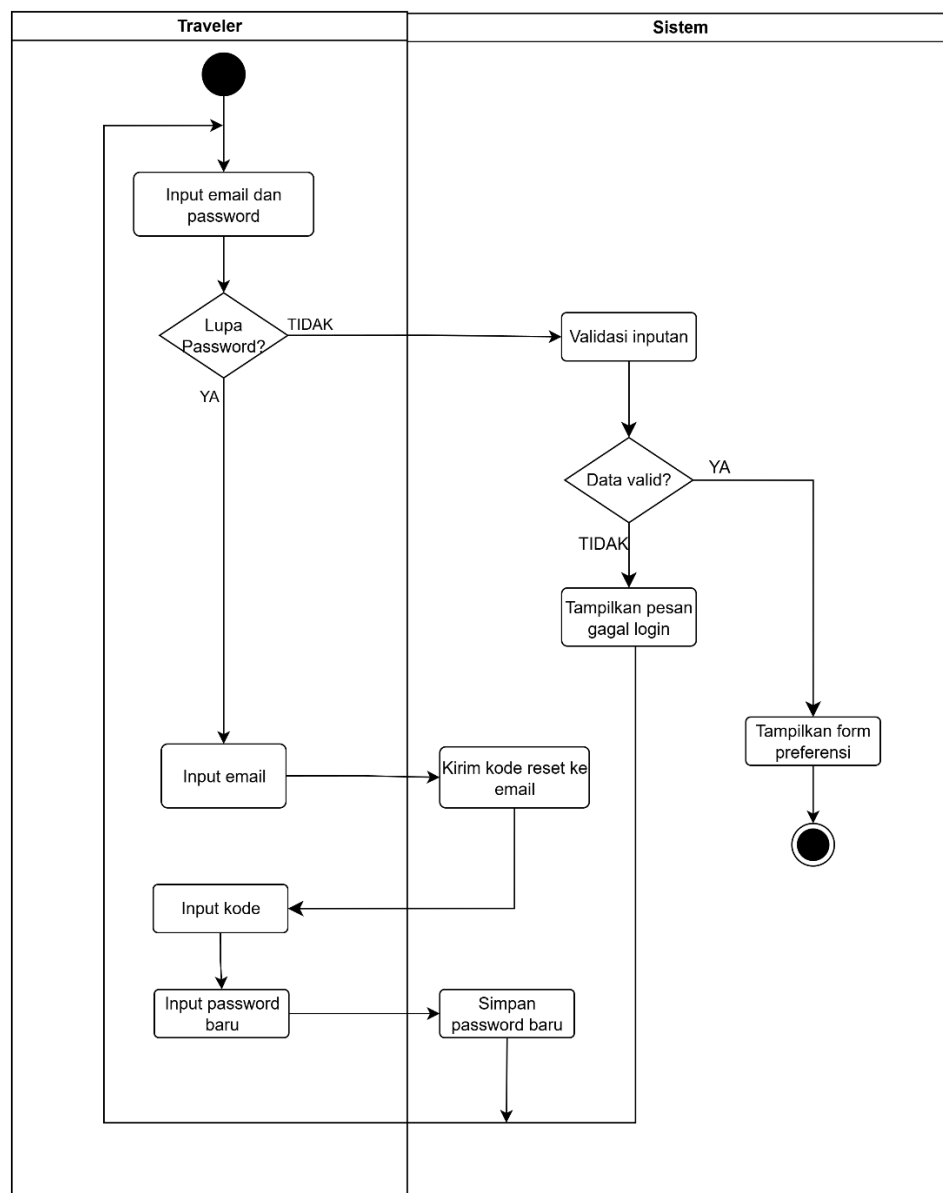
Activity diagram di atas menggambarkan proses registrasi akun oleh pengguna pada sistem Traveloop. Proses diawali ketika traveler mengisi formulir registrasi dengan data seperti nama, email, dan password. Data tersebut kemudian dikirim ke sistem untuk divalidasi.

Validasi ini mencakup pengecekan kelengkapan data, format email yang benar, kekuatan password, dan memastikan bahwa email belum pernah digunakan sebelumnya.

Setelah validasi dilakukan, sistem mengevaluasi apakah data yang dimasukkan valid. Jika data tidak valid, sistem akan menampilkan pesan kesalahan dan pengguna diminta untuk memperbaiki data yang salah dengan kembali ke tahap pengisian formulir.

Jika data valid, sistem akan menyimpan data akun ke dalam basis data. Setelah berhasil disimpan, sistem akan menampilkan notifikasi bahwa pendaftaran telah berhasil. Proses registrasi kemudian selesai.

2. Login

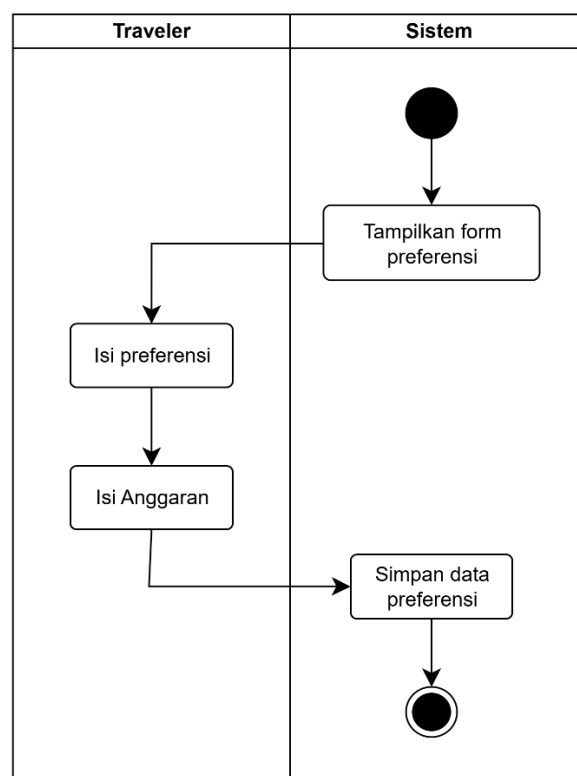


Activity diagram di atas menggambarkan alur login pengguna ke dalam sistem, termasuk penanganan kasus ketika pengguna lupa password.

Proses dimulai ketika traveler membuka halaman login dan menginput email serta password. Setelah data diisi, sistem melakukan validasi terhadap inputan tersebut. Jika data login valid (email dan password sesuai dengan data yang tersimpan), maka sistem akan mengizinkan akses dan menampilkan form preferensi pengguna. Namun, jika data tidak valid (contohnya salah password), sistem akan menampilkan pesan kesalahan bahwa login gagal.

Di halaman login ini juga terdapat opsi "Lupa Password?" yang dapat diklik oleh pengguna jika mereka lupa password-nya. Jika opsi ini dipilih, pengguna akan diarahkan ke proses reset password. Dalam proses tersebut, traveler diminta untuk menginput alamat email yang digunakan saat mendaftar. Sistem kemudian mengirimkan kode verifikasi ke email tersebut. Setelah pengguna menerima dan menginput kode tersebut, mereka dapat membuat password baru. Sistem akan menyimpan password baru yang diinput, dan proses reset password pun selesai. Selanjutnya, pengguna dapat login kembali menggunakan password yang baru saja dibuat.

3. Isi Preferensi



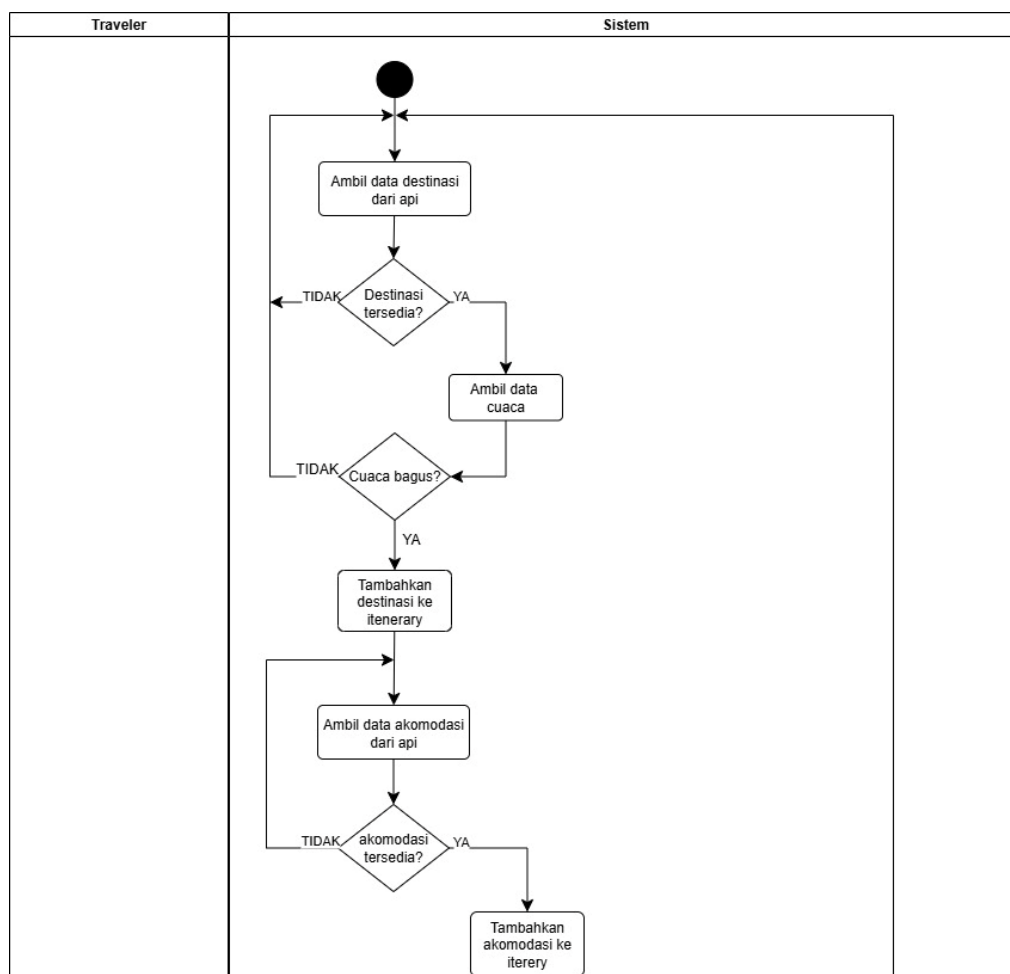
Activity diagram di atas menggambarkan alur aktivitas traveler dalam mengisi preferensi perjalanan pada sistem TravelLoop. Proses dimulai setelah pengguna berhasil login.

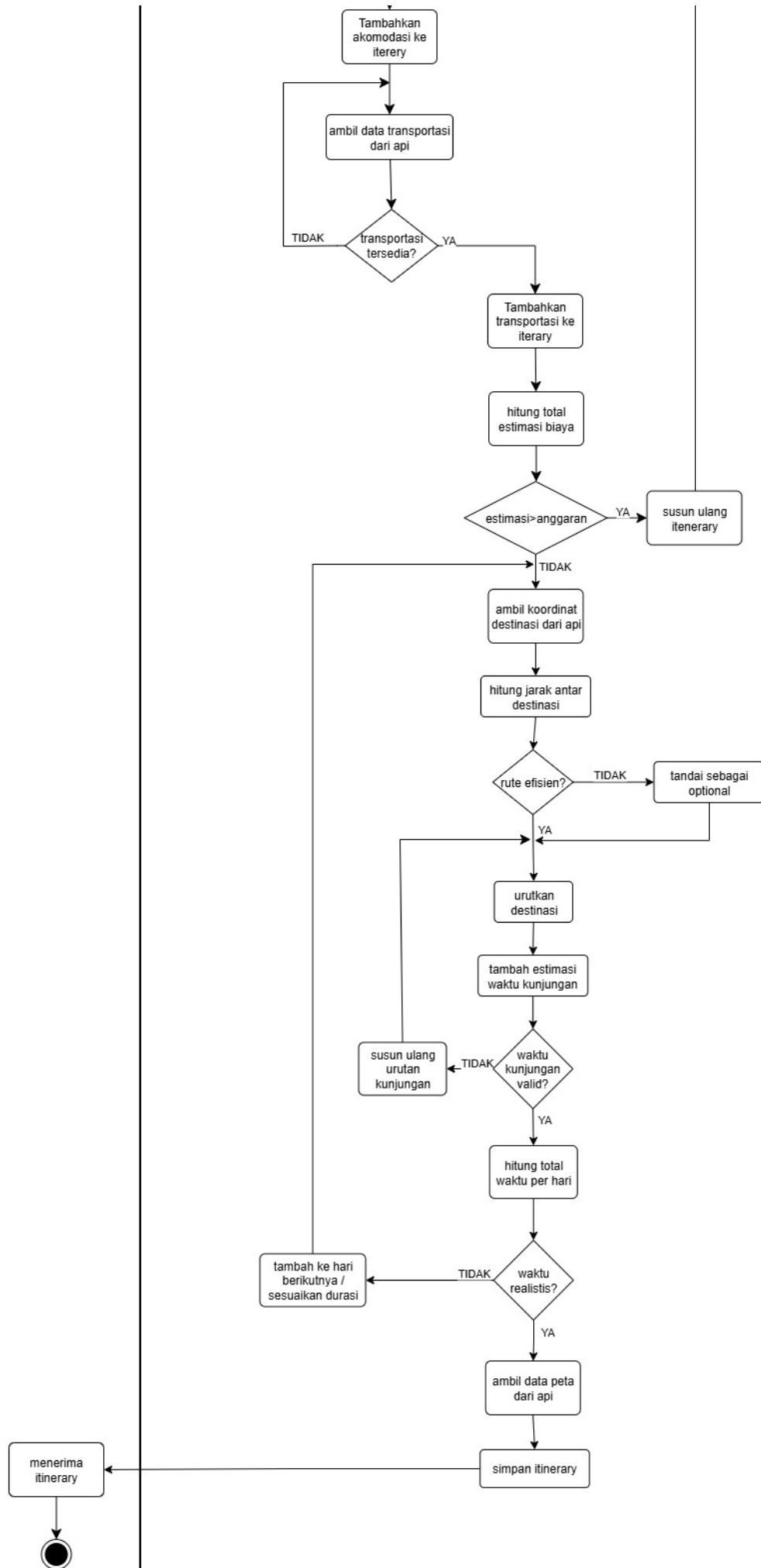
Sistem kemudian menampilkan form pengisian preferensi, yang berisi pilihan terkait rencana perjalanan pengguna. Preferensi ini bisa mencakup berbagai hal seperti jenis destinasi (alam, budaya, kuliner), durasi perjalanan, dan lain-lain.

Selanjutnya, pengguna juga diminta untuk mengisi anggaran perjalanan yang tersedia. Anggaran ini sangat penting karena akan digunakan oleh sistem untuk menyesuaikan rekomendasi destinasi, akomodasi, dan transportasi yang sesuai dengan kemampuan finansial pengguna.

Setelah seluruh data preferensi dan anggaran diisi, sistem akan menyimpan informasi tersebut ke dalam basis data. Informasi ini nantinya digunakan sebagai dasar untuk membentuk itinerary otomatis dan rekomendasi perjalanan yang dipersonalisasi sesuai dengan preferensi dan budget pengguna.

4. Itinerary Otomatis





Activity diagram di atas menggambarkan alur sistem dalam menyusun itinerary perjalanan secara otomatis berdasarkan preferensi dan anggaran yang telah diisi oleh traveler. Proses dimulai dari sistem yang mengambil data destinasi dari API eksternal. Kemudian, sistem memeriksa apakah destinasi tersedia. Jika tidak tersedia, proses mengulang proses mencari destinasi sampai destinasi yang dicari ada atau tersedia. Jika tersedia, sistem melanjutkan dengan mengambil data cuaca dari API dan mengecek apakah cuaca di destinasi tersebut bagus. Jika cuaca tidak mendukung, destinasi tidak dimasukkan ke itinerary. Namun jika cuaca bagus, maka sistem menambahkan destinasi ke daftar itinerary.

Selanjutnya, sistem mengambil data akomodasi dari API dan mengecek ketersediaannya. Jika akomodasi tersedia, maka ditambahkan ke itinerary. Jika tidak, sistem akan mengulangi proses mencari akomodasi sampai ada akomodasi yang sesuai. Sistem kemudian mengambil data transportasi dan juga melakukan pengecekan. Jika tersedia, transportasi ditambahkan ke itinerary. Jika tidak tersedia, sistem akan terus mencari transportasi yang sesuai.

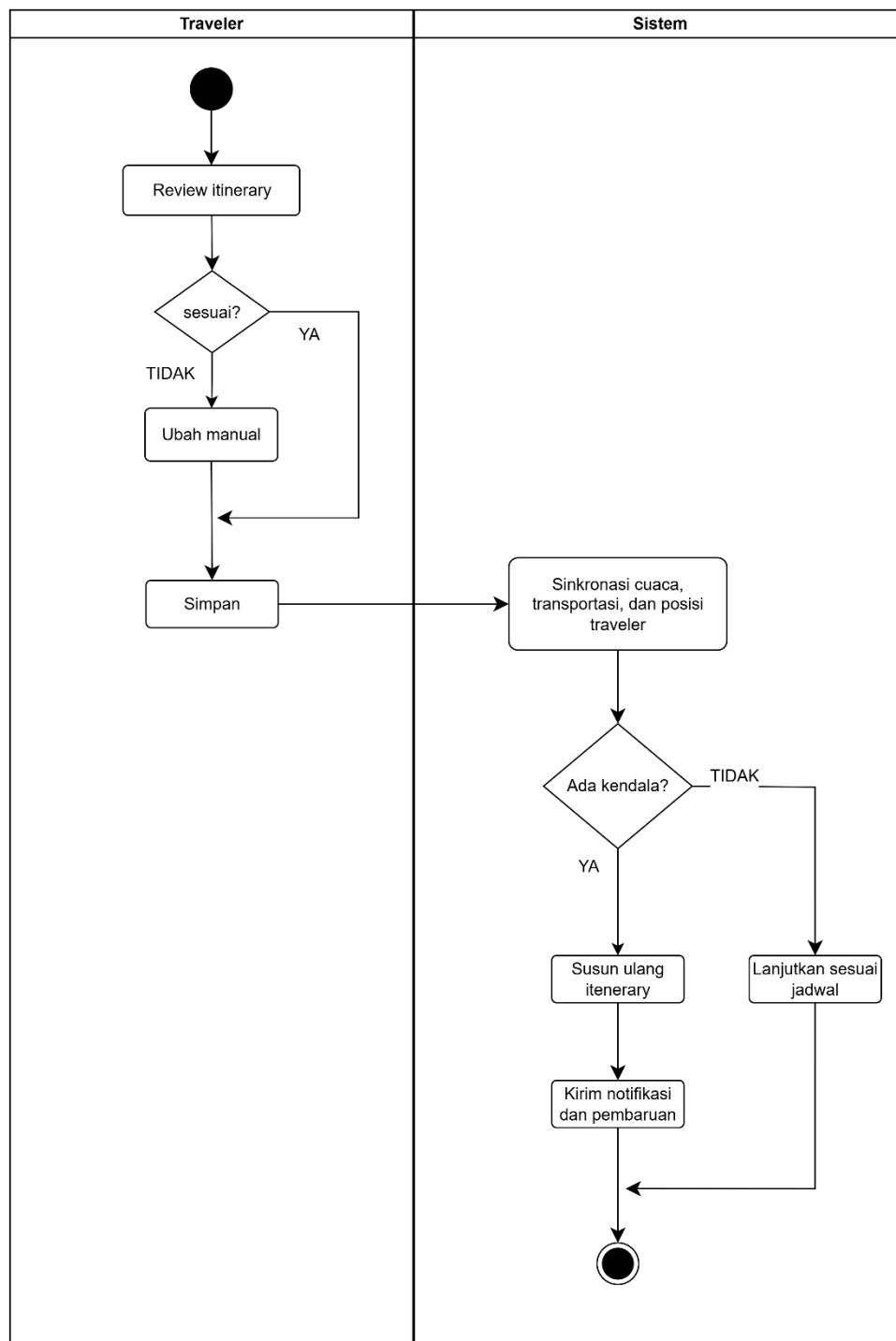
Setelah semua komponen (destinasi, akomodasi, dan transportasi) terkumpul, sistem menghitung total estimasi biaya perjalanan. Jika estimasi melebihi anggaran yang telah ditentukan oleh traveler, maka sistem akan menyusun ulang itinerary (misalnya dengan mengganti atau mengurangi destinasi, akomodasi, atau transportasi). Jika estimasi sesuai anggaran, sistem melanjutkan proses.

Sistem kemudian mengambil koordinat dari masing-masing destinasi untuk menghitung jarak antar destinasi. Jika rute yang didapat tidak efisien, maka destinasi tertentu dapat ditandai sebagai opsional. Jika rute sudah efisien, sistem akan mengurutkan destinasi dan menambahkan estimasi waktu kunjungan.

Selanjutnya, sistem mengevaluasi apakah urutan waktu kunjungan valid dan sesuai dengan jadwal. Jika tidak valid, sistem akan menyusun ulang urutan kunjungan. Jika valid, sistem menghitung total waktu per hari dan mengecek apakah waktu yang tersedia cukup realistis. Jika waktu tidak realistis (terlalu padat), maka kunjungan dapat dipindahkan ke hari berikutnya atau durasi perjalanan akan disesuaikan lagi.

Setelah semua pengecekan selesai dan waktu realistis, sistem mengambil data peta dari API untuk mendukung navigasi perjalanan. Itinerary yang telah disusun secara otomatis disimpan ke sistem dan traveler menerima hasil itinerary tersebut.

5. Validasi Itinerary



Activity diagram validasi itinerary dimulai dari sisi traveler, di mana pengguna terlebih dahulu melakukan review itinerary yang telah disusun oleh sistem. Setelah meninjau, traveler akan menentukan apakah itinerary tersebut sudah sesuai dengan preferensi dan kebutuhan mereka. Jika belum sesuai, traveler diberi opsi untuk melakukan perubahan secara manual (misalnya mengganti destinasi, mengubah waktu kunjungan, atau menyesuaikan urutan

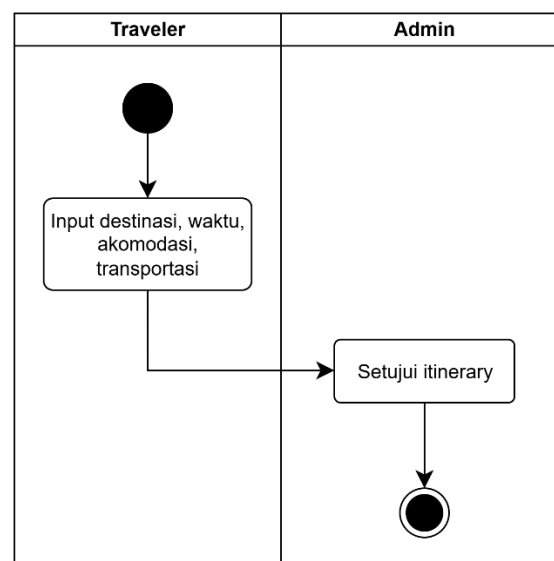
kegiatan). Setelah perubahan selesai, pengguna menyimpan itinerary hasil modifikasi tersebut.

Selanjutnya, sistem akan menjalankan proses sinkronisasi data, yang mencakup pembaruan informasi cuaca, ketersediaan transportasi, dan posisi lokasi traveler secara real-time. Tujuannya adalah untuk memastikan itinerary tetap relevan dengan kondisi terkini di lapangan.

Setelah sinkronisasi dilakukan, sistem akan memeriksa apakah terdapat kendala yang dapat mengganggu jalannya itinerary, seperti cuaca buruk, keterlambatan transportasi, atau lokasi traveler terlalu jauh dari destinasi selanjutnya. Jika tidak ada kendala, itinerary tetap berjalan sesuai jadwal.

Namun, jika ditemukan kendala, sistem akan secara otomatis melakukan penyusunan ulang itinerary dengan menyesuaikan urutan atau mengganti aktivitas tertentu agar tetap optimal. Setelah disusun ulang, sistem akan mengirimkan notifikasi dan pembaruan itinerary kepada traveler agar mereka mengetahui perubahan yang terjadi.

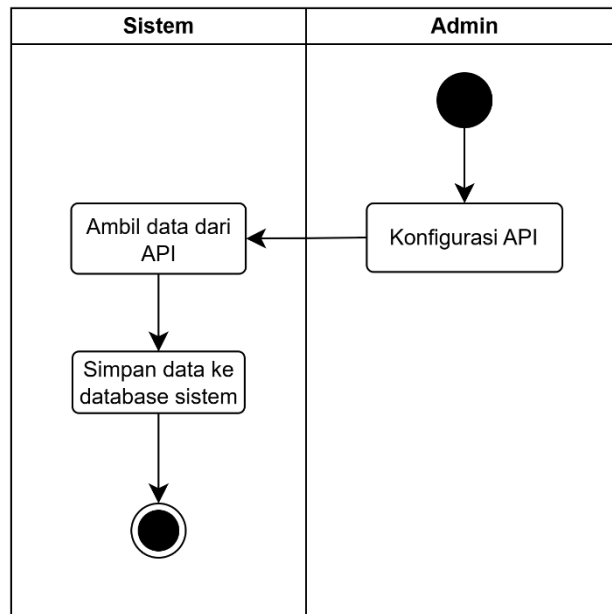
6. Itinerary Custom



Activity diagram ini menggambarkan alur aktivitas ketika traveler membuat itinerary secara manual (custom) sesuai keinginan mereka, dan kemudian disimpan oleh admin dalam sistem TravelLoop. Proses dimulai dari sisi traveler, di mana pengguna mengisi data itinerary secara manual. Data yang dimasukkan mencakup destinasi, waktu kunjungan, akomodasi, dan transportasi yang diinginkan. Proses ini memungkinkan traveler untuk menentukan sendiri seluruh elemen perjalanan tanpa mengikuti sistem itinerary otomatis.

Setelah data itinerary custom diinput oleh traveler, informasi tersebut dikirimkan ke sistem dan diterima oleh pihak admin. Admin kemudian melakukan proses penyimpanan itinerary custom ke dalam basis data sistem, agar dapat digunakan atau diakses kembali di kemudian hari oleh traveler tersebut. Proses berakhir setelah data itinerary berhasil disetujui admin.

7. API



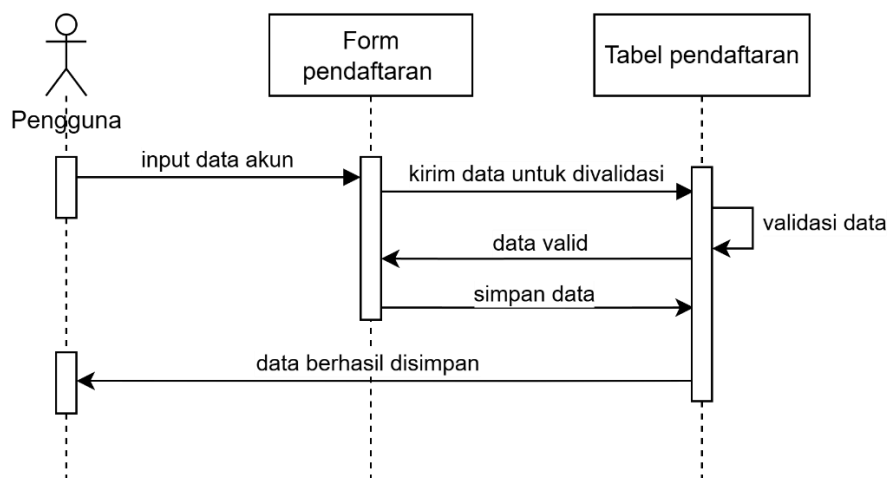
Activity diagram ini menggambarkan alur proses ketika sistem TraveLoop mengambil data dari API eksternal, seperti API cuaca, transportasi, atau destinasi wisata, untuk mendukung penyusunan itinerary perjalanan. Proses dimulai dari peran admin, yang bertugas melakukan konfigurasi API. Konfigurasi ini mencakup pengaturan endpoint, API key, parameter permintaan, dan frekuensi pembaruan data. Konfigurasi ini penting agar sistem dapat terhubung dengan benar ke penyedia data eksternal.

Setelah API dikonfigurasi, sistem secara otomatis melakukan proses pengambilan data dari API sesuai konfigurasi yang telah ditentukan. Data yang diambil bisa berupa informasi cuaca terkini, daftar transportasi yang tersedia, atau detail destinasi wisata yang up-to-date. Selanjutnya, setelah data berhasil diambil, sistem akan menjalankan proses penyimpanan data ke dalam database internal. Tujuannya adalah agar data dari API eksternal dapat digunakan secara efisien oleh berbagai fitur dalam sistem, seperti itinerary otomatis, rekomendasi destinasi, dan validasi kondisi lapangan secara real-time. Proses ini berakhir ketika seluruh data yang diperlukan dari API telah berhasil disimpan di sistem.

V. Sequence Diagram

Sequence diagram pada sistem TraveLoop digunakan untuk menggambarkan alur interaksi antara pengguna dan komponen sistem. Mulai dari registrasi, login, pengisian preferensi, pembuatan itinerary otomatis maupun custom, hingga pengelolaan konfigurasi API. Penjelasan berikut disusun untuk mempermudah pemahaman terhadap alur kerja sistem berdasarkan skenario yang terjadi.

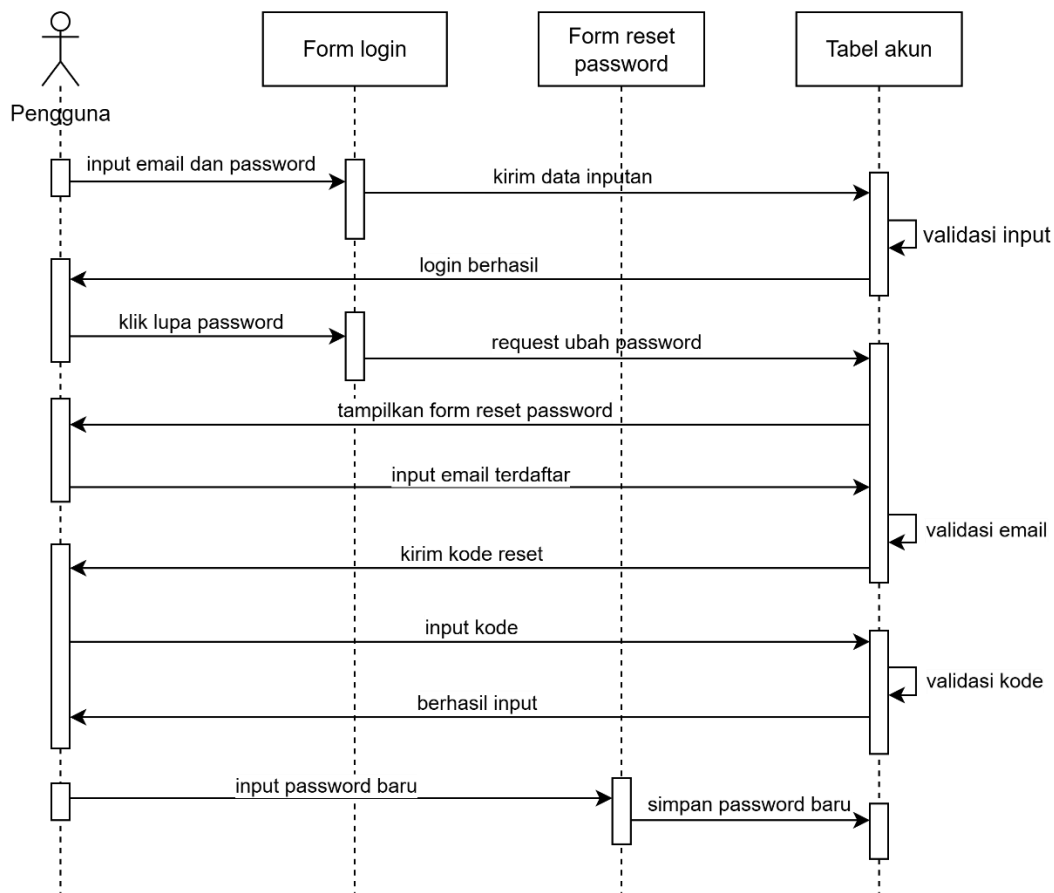
1. Registrasi



Sequence diagram proses registrasi menggambarkan alur interaksi antara pengguna, form pendaftaran, dan tabel pendaftaran (sebagai representasi database). Proses dimulai ketika pengguna mengisi data akun, seperti nama, email, dan password melalui form pendaftaran. Setelah data diinput, form pendaftaran akan mengirimkan informasi tersebut ke tabel pendaftaran untuk dilakukan validasi. Validasi bertujuan untuk memastikan bahwa data yang dimasukkan sesuai format yang benar dan belum pernah terdaftar sebelumnya.

Jika data dinyatakan valid, form pendaftaran akan melanjutkan proses dengan menginstruksikan sistem untuk menyimpan data tersebut ke dalam database. Setelah data berhasil disimpan, sistem memberikan notifikasi kepada traveler bahwa proses pendaftaran telah berhasil.

2. Login



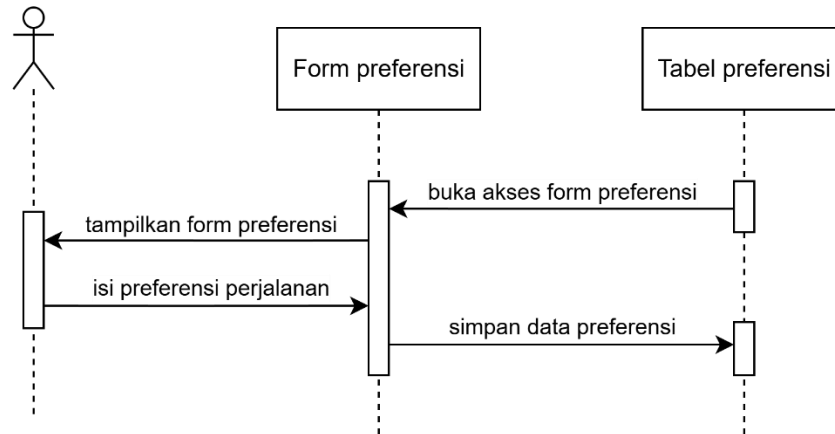
Sequence diagram ini menggambarkan alur interaksi antara pengguna, form login, form reset password, dan tabel akun dalam proses login dan pemulihan akun di sistem. Proses dimulai saat pengguna menginput email dan password ke form login. Form login kemudian mengirimkan data inputan tersebut ke sistem untuk diverifikasi melalui tabel akun. Jika data valid, maka login berhasil dan akses diberikan kepada pengguna.

Namun, jika pengguna lupa password, ia dapat menekan opsi "lupa password" pada form login. Aksi ini mengaktifkan form reset password, yang kemudian menampilkan halaman untuk memulai proses pemulihan. Pengguna diminta mengisi email yang terdaftar di sistem, lalu form reset password mengirimkan permintaan ke tabel akun untuk memvalidasi keberadaan email tersebut.

Jika email valid, sistem mengirimkan kode verifikasi (reset code) ke email pengguna. Selanjutnya, pengguna memasukkan kode tersebut ke dalam form reset password. Kode ini kemudian diverifikasi oleh sistem. Jika validasi berhasil, pengguna dapat menginput

password baru. Langkah terakhir adalah menyimpan password baru ke dalam tabel akun sebagai pengganti password lama.

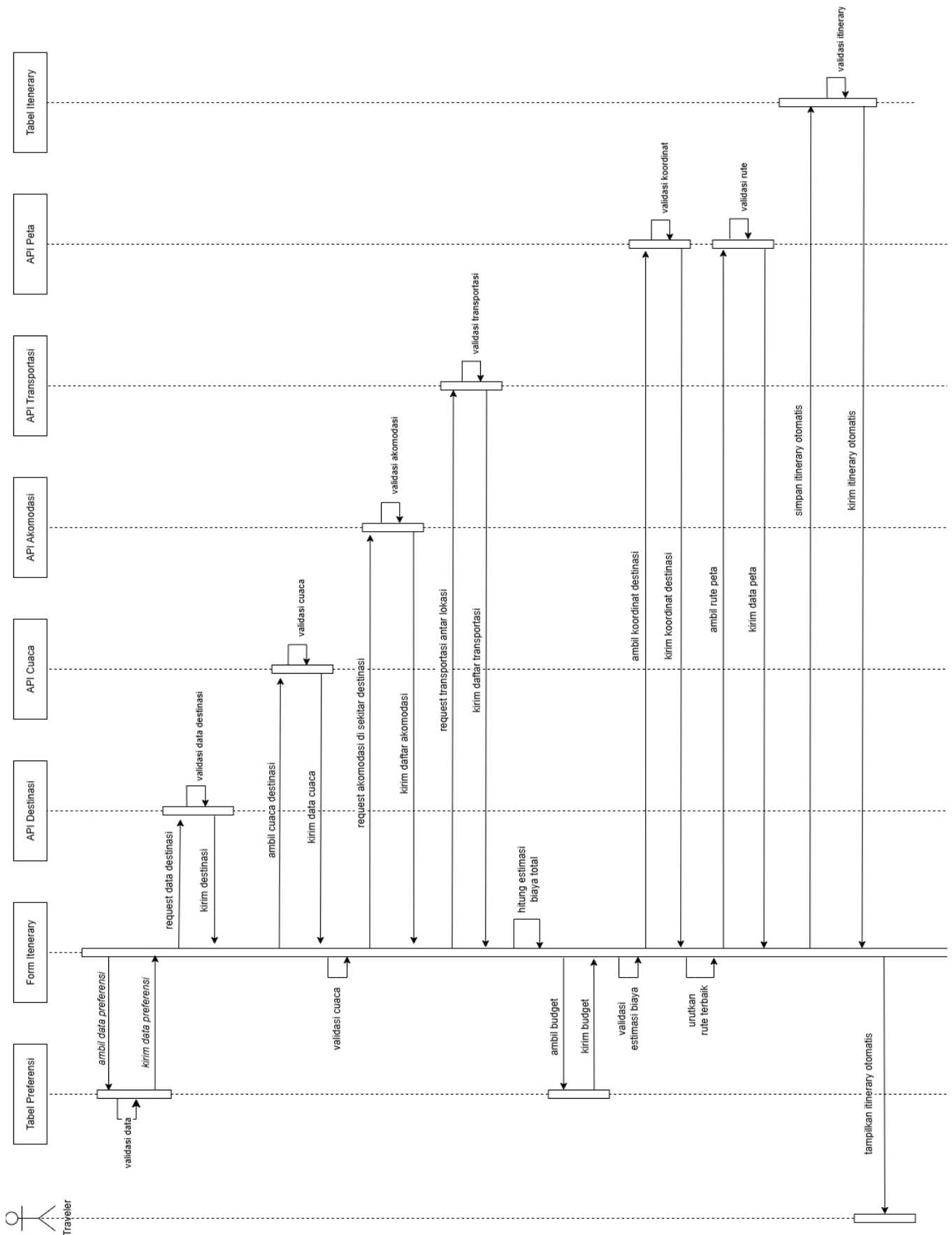
3. Isi Preferensi



Sequence diagram ini menggambarkan proses interaksi antara pengguna (Traveler), Form Preferensi, dan Tabel Preferensi dalam pengisian dan penyimpanan data preferensi di sistem Traveloop. Proses dimulai ketika traveler mengakses form preferensi yang ditampilkan oleh sistem. Traveler kemudian mengisi preferensi perjalanan seperti jenis wisata, tujuan, kategori minat, dan anggaran yang diinginkan.

Setelah traveler menginput data preferensi, Form Preferensi mengirimkan permintaan akses ke Tabel Preferensi untuk memastikan koneksi penyimpanan data tersedia. Setelah akses diberikan, Form Preferensi melanjutkan dengan mengirimkan seluruh data preferensi yang telah diisi ke Tabel Preferensi. Tabel Preferensi kemudian memproses dan menyimpan informasi tersebut ke dalam sistem penyimpanan (database). Proses ini berakhir ketika data berhasil disimpan, menandai bahwa preferensi traveler telah tercatat dengan sukses dalam sistem Traveloop.

4. Itinerary Otomatis



Sequence diagram ini menjelaskan alur pembuatan itinerary otomatis pada sistem Traveloop. Proses diawali saat traveler mengambil data preferensi dari Tabel Preferensi. Setelah data preferensi diambil dan divalidasi, informasi tersebut dikirimkan ke Form Itinerary sebagai dasar pembuatan itinerary.

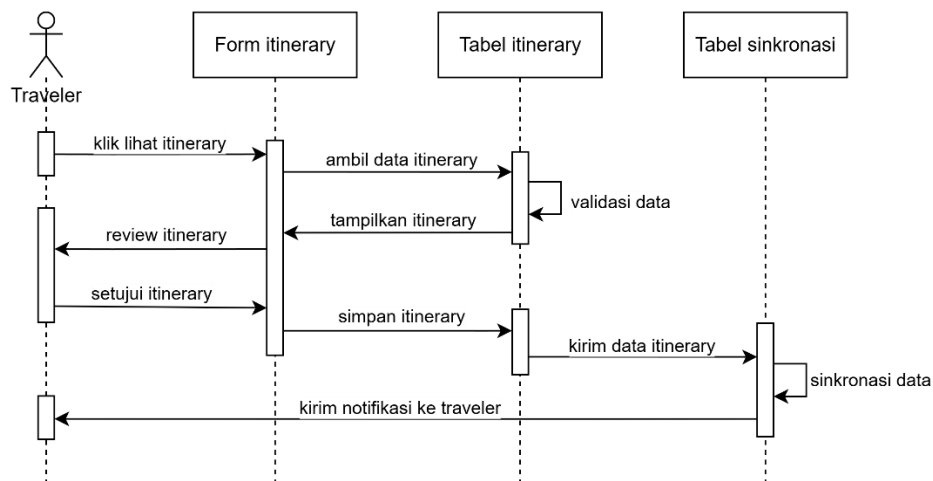
Form Itinerary kemudian melakukan permintaan data destinasi ke API Destinasi. Setelah data destinasi divalidasi dan dikirim, Form Itinerary melanjutkan dengan mengambil informasi cuaca dari API Cuaca untuk memastikan kondisi destinasi mendukung. Setelah data cuaca diterima dan divalidasi, sistem melakukan pencarian akomodasi di sekitar destinasi melalui API Akomodasi. Hasil pencarian tersebut dikirimkan kembali ke Form Itinerary setelah validasi.

Langkah selanjutnya adalah permintaan transportasi antar lokasi melalui API Transportasi. Setelah daftar transportasi divalidasi dan dikirim ke Form Itinerary, sistem menghitung estimasi total biaya dari keseluruhan itinerary. Untuk memastikan estimasi tersebut sesuai anggaran, Form Itinerary mengambil data budget dari Tabel Preferensi dan melakukan proses validasi terhadap estimasi biaya tersebut.

Jika estimasi dianggap sesuai, sistem mengirim permintaan ke API Peta untuk mendapatkan koordinat geografis dari destinasi. Setelah koordinat diterima dan divalidasi, Form Itinerary melakukan pengurutan rute terbaik. Sistem juga mengambil rute peta dari API Peta, lalu memvalidasi dan mengirimkan data rute kembali ke Form Itinerary.

Setelah itinerary terbentuk sepenuhnya, Form Itinerary mengirimkan data untuk disimpan ke Tabel Itinerary. Tabel Itinerary akan memvalidasi data tersebut, lalu mengirim kembali salinan itinerary yang telah disimpan. Sebagai tahap akhir, Form Itinerary menampilkan itinerary otomatis kepada traveler sebagai hasil dari seluruh proses perencanaan perjalanan yang dilakukan secara otomatis oleh sistem.

5. Validasi Itinerary



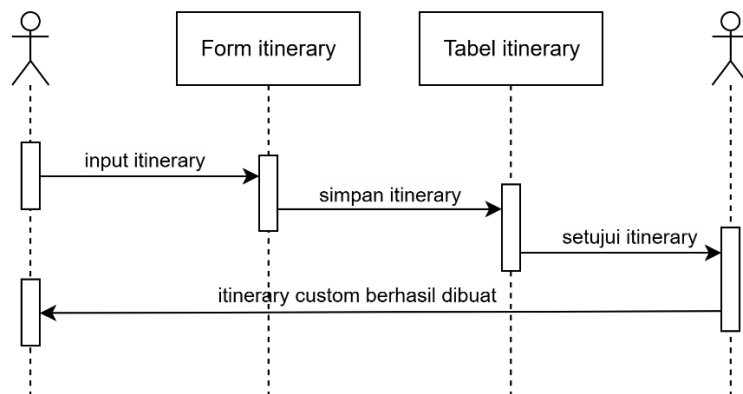
Sequence diagram ini menggambarkan alur interaksi antara traveler, Form Itinerary, Tabel Itinerary, dan Tabel Sinkronisasi dalam proses mereview itinerary yang telah dibuat sistem. Proses dimulai saat pengguna menekan tombol untuk melihat itinerary. Permintaan ini dikirim ke Form Itinerary, lalu diteruskan ke Tabel Itinerary untuk mengambil data itinerary yang dimaksud.

Tabel Itinerary kemudian melakukan validasi untuk memastikan data yang diambil benar dan masih relevan. Setelah divalidasi, data itinerary dikirim kembali ke Form Itinerary dan ditampilkan kepada pengguna untuk ditinjau. Setelah melakukan review, pengguna menyetujui itinerary tersebut dengan mengirimkan persetujuan ke Form Itinerary.

Berdasarkan persetujuan itu, Form Itinerary menyimpan atau memperbarui data itinerary ke dalam Tabel Itinerary. Selanjutnya, Tabel Itinerary mengirimkan data itinerary ke Tabel Sinkronisasi. Tabel Sinkronisasi kemudian melakukan proses sinkronisasi agar data itinerary terbaru tersedia secara konsisten di seluruh sistem atau layanan terhubung.

Sebagai langkah akhir, Form Itinerary mengirimkan notifikasi kepada traveler untuk menginformasikan bahwa itinerary telah tersimpan dengan sukses. Proses ini memastikan itinerary yang dibuat telah disetujui oleh pengguna, serta disinkronkan dengan sistem internal Traveloop secara menyeluruh.

6. Itinerary Custom



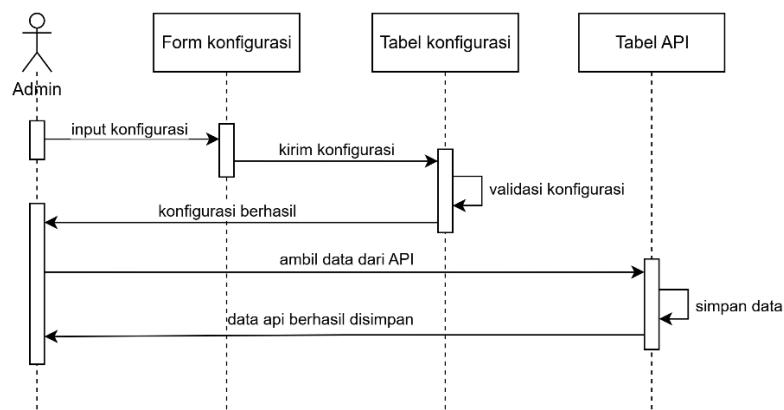
Sequence diagram ini menggambarkan proses pembuatan itinerary secara custom oleh traveler. Proses dimulai ketika pengguna mengisi detail itinerary yang diinginkan melalui Form Itinerary. Data yang dimasukkan mencakup destinasi, tanggal perjalanan, akomodasi, transportasi, dan lain lain.

Setelah seluruh data itinerary custom diisi, Form Itinerary mengirimkan permintaan untuk menyimpan data tersebut ke Tabel Itinerary. Tabel Itinerary kemudian menjadi aktif untuk menerima dan menyimpan data yang baru dibuat oleh pengguna ke dalam sistem.

Setelah penyimpanan berhasil, sistem secara otomatis menyetujui itinerary tersebut. Persetujuan ini dilakukan langsung oleh Tabel Itinerary, yang mengindikasikan bahwa itinerary custom dianggap sah tanpa perlu verifikasi tambahan, karena dibuat langsung oleh pengguna sendiri.

Sebagai langkah akhir, Form Itinerary mengirimkan notifikasi kepada pengguna bahwa itinerary custom telah berhasil dibuat. Menandakan bahwa itinerary telah tersimpan dan siap digunakan dalam sistem Traveloop.

7. API



Sequence diagram ini menggambarkan alur interaksi antara admin, Form Konfigurasi, Tabel Konfigurasi, dan Tabel API dalam proses input dan penyimpanan konfigurasi API eksternal pada sistem Traveloop. Proses dimulai saat admin mengisi detail konfigurasi API melalui Form Konfigurasi. Detail ini bisa berupa endpoint, metode request, API key, atau parameter lain yang dibutuhkan untuk mengakses data eksternal.

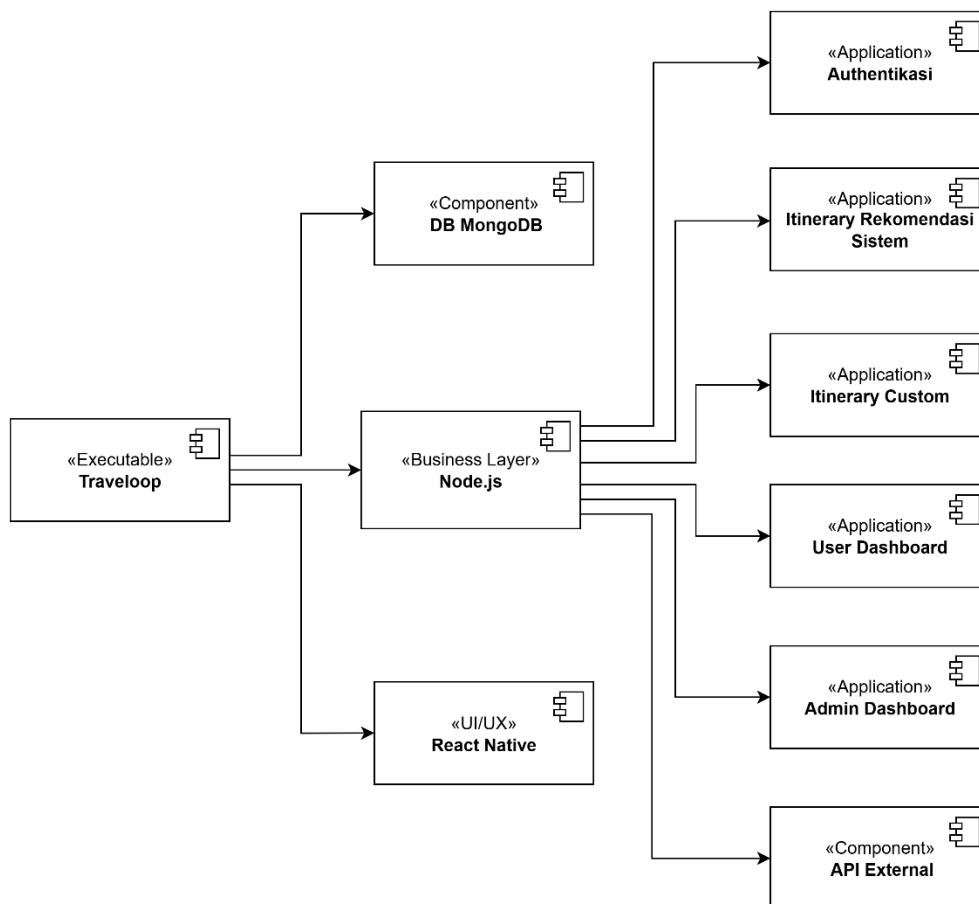
Setelah admin selesai mengisi data, Form Konfigurasi mengirimkan konfigurasi tersebut ke Tabel Konfigurasi untuk diproses. Tabel Konfigurasi akan melakukan validasi terhadap data yang diterima, memastikan bahwa konfigurasi yang dimasukkan telah lengkap dan sesuai format yang dibutuhkan sistem.

Jika validasi berhasil, Form Konfigurasi mengirimkan notifikasi kepada admin bahwa konfigurasi telah berhasil disimpan. Setelah itu, Tabel Konfigurasi secara otomatis memulai proses pengambilan data dari sumber API eksternal sesuai dengan konfigurasi yang baru disimpan, dan mengirimkan permintaan ke Tabel API.

Setelah mengambil data dari API data akan disimpan ke dalam sistem. Sebagai langkah akhir, sistem memberi informasi bahwa data dari API telah berhasil disimpan, memastikan bahwa informasi dari sumber eksternal kini tersedia di dalam Traveloop dan siap digunakan oleh fitur-fitur lain dalam sistem.

VI. Component Diagram

Component diagram sistem Traveloop menggambarkan struktur modular aplikasi serta hubungan antar komponen utama yang membentuk keseluruhan sistem. Diagram ini menekankan bagaimana bagian-bagian seperti antarmuka pengguna, logika bisnis, database, dan layanan eksternal saling terhubung dan berinteraksi untuk mendukung fungsionalitas aplikasi.



Komponen Utama:

1. «Executable» Traveloop

Merupakan aplikasi utama yang dijalankan pengguna. Komponen ini menjadi titik masuk utama bagi seluruh interaksi dengan sistem, dan menggabungkan antarmuka serta logika aplikasi dalam satu kesatuan.

2. «UI/UX» React Native

Berfungsi sebagai antarmuka pengguna yang dibangun dengan React Native. Komponen ini menangani tampilan visual dan interaksi pengguna, serta mengirimkan permintaan ke lapisan bisnis untuk memproses data.

3. «Business Layer» Node.js

Merupakan inti dari sistem, tempat seluruh logika bisnis dijalankan. Komponen ini mengatur alur data antara antarmuka, database, dan layanan eksternal. Node.js juga menangani komunikasi antar modul dan memastikan integritas proses.

4. «Component» DB MongoDB

Basis data MongoDB yang digunakan untuk menyimpan seluruh data sistem, seperti akun pengguna, data itinerary. Komunikasi hanya dilakukan melalui lapisan bisnis untuk menjaga struktur sistem tetap terkontrol.

Komponen Pendukung:

- «Application» Authentikasi
Menangani proses login, registrasi, dan verifikasi identitas pengguna.
- «Application» Itinerary Rekomendasi Sistem
Mengelola pembentukan itinerary otomatis berdasarkan preferensi pengguna.
- «Application» Itinerary Custom
Mendukung pembuatan itinerary manual oleh pengguna secara fleksibel.
- «Application» User Dashboard
Menyediakan fitur utama dan navigasi khusus bagi pengguna umum.
- «Application» Admin Dashboard
Menyediakan antarmuka dan alat manajemen bagi admin dalam mengelola sistem dan data.
- «Component» API External
Meliputi integrasi dengan layanan pihak ketiga seperti API cuaca, peta, dan transportasi. Lapisan bisnis bertugas memanggil dan memproses data dari API ini untuk digunakan dalam sistem.

Relasi Antar Komponen:

Komponen Traveloop sebagai executable bergantung pada UI/UX React Native dan Business Layer Node.js. Antarmuka pengguna mengirimkan permintaan ke Node.js, yang kemudian memproses permintaan tersebut dengan bantuan database MongoDB dan API eksternal. Seluruh modul aplikasi (Authentikasi, Itinerary, Dashboard) juga berada di bawah koordinasi langsung dari lapisan bisnis ini.