

## BAB III

### ANALISA DAN PERANCANGAN

#### 3.1 Analisa

##### 3.1.1 Identifikasi Masalah

Keterbatasan akses informasi saat *offline* menjadi kendala besar bagi wisatawan. Dari wawancara dengan wisatawan mengungkapkan kesulitan mereka dalam mengakses informasi penting seperti panduan perjalanan atau *travel blog* saat berada di lokasi tanpa koneksi internet. Bahkan beberapa wisatawan sengaja mematikan koneksi internet untuk menghemat kuota atau baterai, meskipun informasi tersebut sangat diperlukan selama aktivitas wisata.

Fragmentasi layanan dan kurangnya integrasi dalam manajemen perjalanan juga menjadi masalah serius. Pada wawancara dengan anggota komunitas Belantara.co menunjukkan bahwa wisatawan terpaksa menggunakan beberapa aplikasi atau metode berbeda untuk merencanakan perjalanan, menyimpan informasi, dan mencari panduan destinasi. Observasi lebih lanjut mengungkapkan bahwa manajemen dan pencatatan informasi perjalanan masih dilakukan secara terpisah, baik melalui catatan manual atau aplikasi digital yang berbeda-beda. Keberagaman metode ini menciptakan kompleksitas tambahan dalam pengelolaan perjalanan, terutama bagi wisatawan dengan mobilitas tinggi.

Keterbatasan kompatibilitas lintas *platform* menjadi hambatan tambahan dalam penggunaan aplikasi wisata. Observasi terhadap aplikasi yang ada menunjukkan bahwa beberapa dari aplikasi tersebut tidak dirancang untuk berfungsi secara optimal di berbagai seperti laptop, ponsel, dan tablet. Akibatnya,

pengalaman pengguna menjadi tidak konsisten dan terbatas. Wawancara dengan wisatawan juga mengungkapkan frustrasi mereka ketika harus beradaptasi dengan antarmuka dan fungsionalitas yang berbeda tergantung pada perangkat yang mereka gunakan, sehingga menambah kesulitan dalam perencanaan dan pelaksanaan perjalanan.

### 3.1.2 Pemecahan Masalah

Berdasarkan identifikasi masalah di atas, dapat diberikan usulan untuk memecahkan masalah tersebut yaitu:

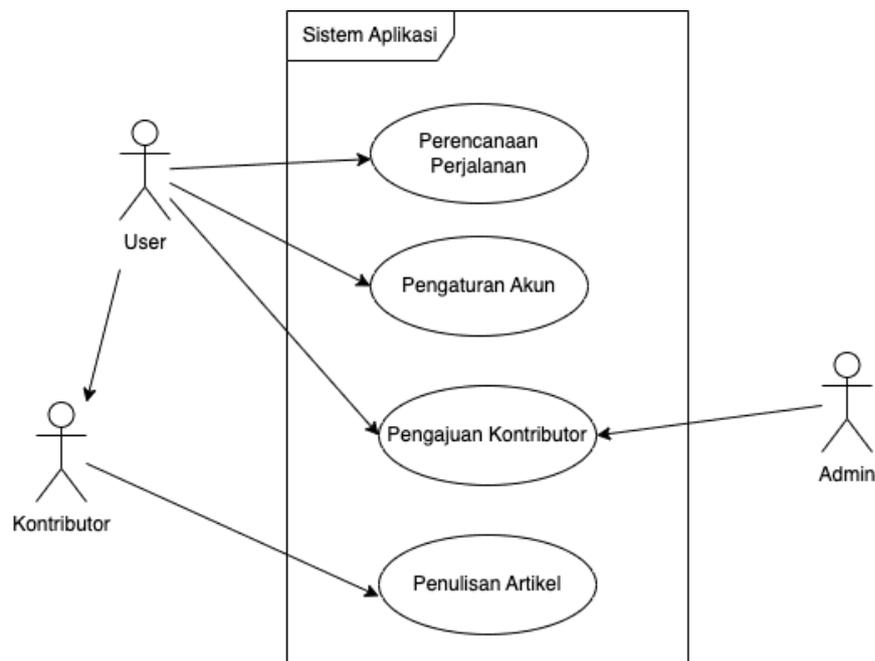
1. Dibuatnya *travel blog* dengan *channel* dari komunitas atau *personal blogger* yang terverifikasi sehingga kualitasnya terjaga yang dapat diakses secara *offline* dengan memanfaatkan fitur dari *progressive web apps*.
2. Dibuatnya fitur rencana perjalanan untuk mendata kebutuhan perjalanan, catatan, dan lokasi destinasi. Fitur ini tentunya juga didukung fungsionalitas tanpa koneksi internet.
3. Rencana perjalanan dapat dipublikasikan secara umum untuk menjadi konten guna menjadi rencana perjalanan yang dapat dibaca wisatawan/pengguna lain.

## 3.2 Perancangan

### 3.2.1 Perancangan Sistem

#### 1. Use Case Diagram

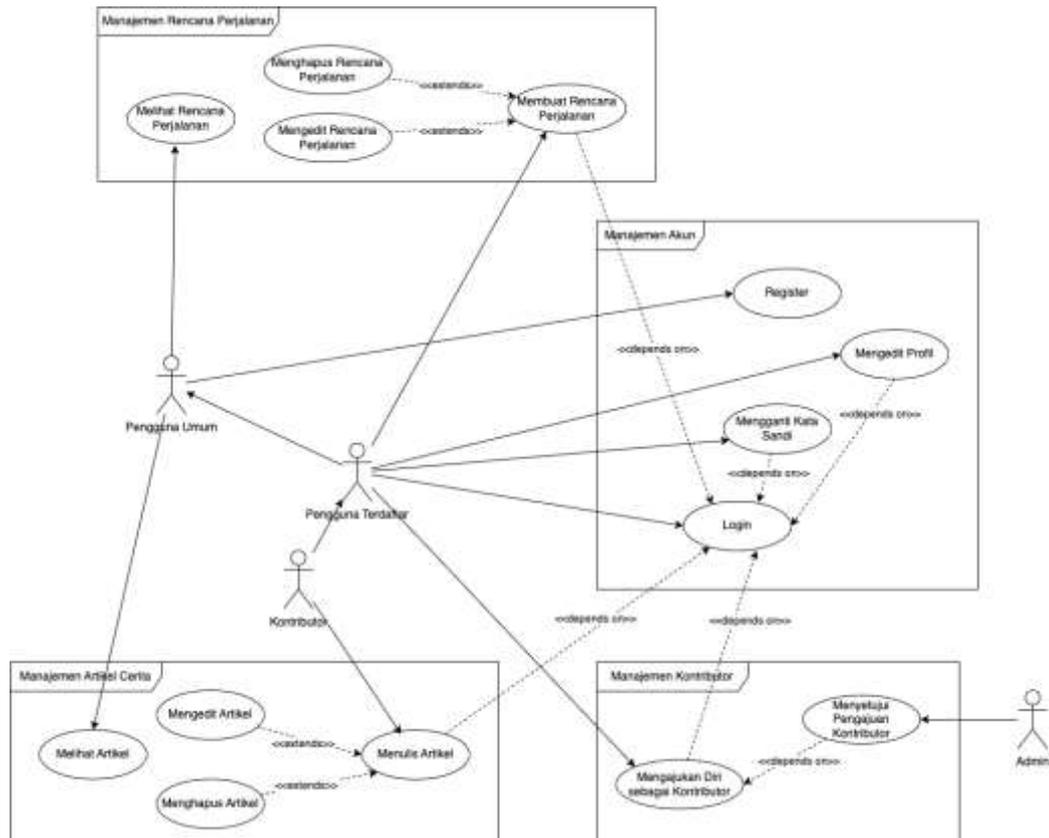
*Use case* diagram adalah representasi visual yang menggambarkan berbagai skenario interaksi antara pengguna dan sistem. Diagram ini berfungsi untuk menunjukkan hubungan antara aktor—yaitu entitas eksternal yang berinteraksi dengan sistem—dan serangkaian tindakan atau kegiatan yang dapat dilakukan oleh aktor tersebut dalam konteks sistem. Dengan demikian, diagram *use case* memberikan gambaran yang jelas mengenai bagaimana pengguna berinteraksi dengan sistem dan apa saja fungsi atau fitur yang tersedia untuk mereka dalam sistem tersebut.



Gambar 3. 1 *Use Case Diagram General*

Pada *use case diagram general* gambar 3.1 menggambarkan interaksi antara aktor-aktor utama (*User*, *Kontributor*, dan *Admin*) dengan sistem aplikasi wisata.

*User*, yang mewakili pengguna umum, memiliki akses ke perencanaan perjalanan untuk membuat dan mengelola rencana perjalanan, pengaturan akun untuk autentikasi dan mengelola akun, dan pengajuan kontributor untuk menambah hak akses mereka dalam sistem. Kontributor, sebagai pengguna dengan hak akses lebih luas, dapat menggunakan fitur penulisan artikel, memungkinkan mereka berbagi pengalaman dan informasi wisata dengan menulis artikel cerita. Admin berperan penting dalam *use case* pengajuan kontributor, menunjukkan adanya proses verifikasi untuk menjamin kualitas konten. Struktur ini mencerminkan sebuah sistem yang dirancang untuk memfasilitasi perencanaan perjalanan, berbagi informasi, dan manajemen konten *user-generated*, dengan mekanisme kontrol yang memastikan kualitas informasi dalam *platform*.



Gambar 3. 2 Use Case Diagram Revised

Pada diagram gambar 3.2 menunjukkan hubungan yang saling terkait antara empat area utama: Manajemen Rencana Perjalanan, Manajemen Akun, Manajemen Artikel Cerita, dan Manajemen Kontributor. Dalam Manajemen Rencana Perjalanan, pengguna dapat melakukan berbagai aktivitas terkait perencanaan perjalanan. Aktivitas ini seperti melihat rencana perjalanan yang tersedia, membuat rencana perjalanan baru, mengedit rencana yang sudah ada, dan menghapus rencana. Fitur-fitur ini saling berhubungan. Manajemen Akun terhubung erat dengan fitur-fitur lainnya, di mana proses registrasi dan *login* menjadi prasyarat untuk mengakses fitur-fitur tertentu seperti membuat rencana perjalanan atau menulis artikel. Pengguna terdaftar juga dapat mengedit profil dan mengganti kata sandi mereka, yang merupakan bagian dari manajemen akun.

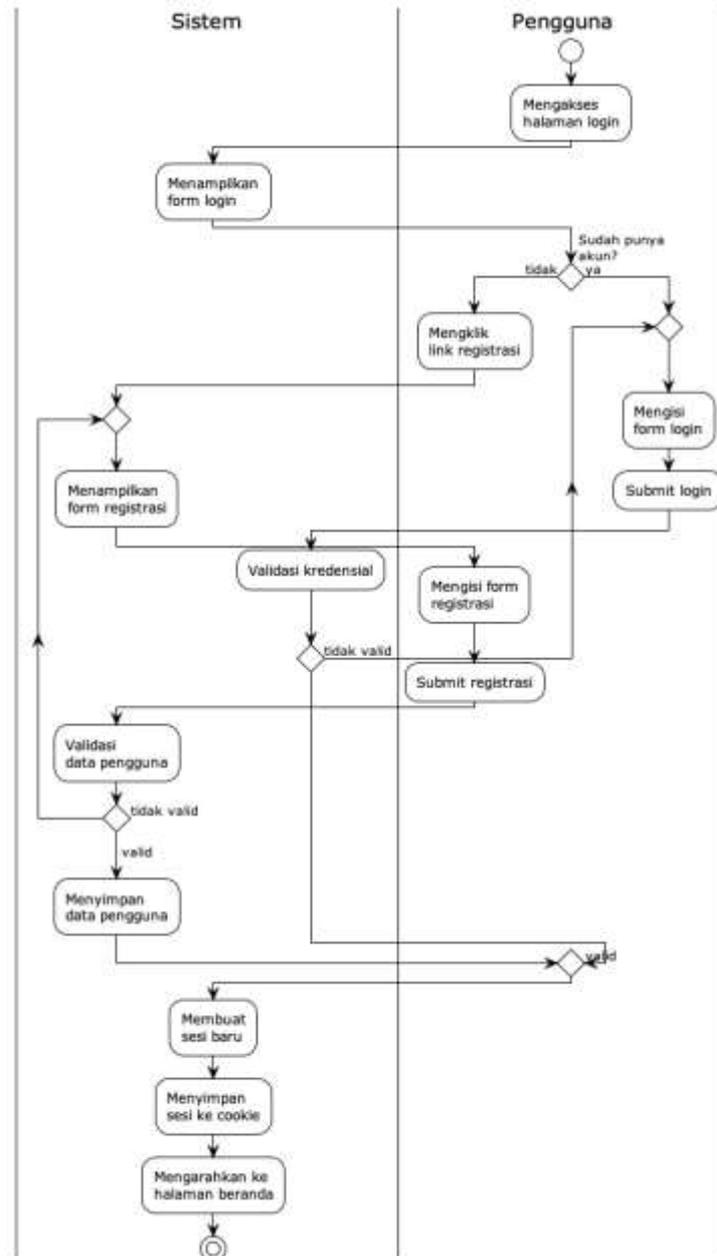


yang lebih luas dan fungsional. Mereka dapat membuat rencana perjalanan baru, yang melibatkan serangkaian proses seperti mengisi informasi dasar perjalanan, mencari dan menambahkan destinasi, mengunggah gambar *cover*, serta menambahkan catatan untuk setiap destinasi.

Selain kemampuan untuk membuat rencana baru, pengguna terdaftar juga dapat mengedit dan menghapus rencana perjalanan yang sudah pernah dibuat. Penting untuk diketahui bahwa semua aktivitas yang dilakukan oleh pengguna terdaftar memerlukan proses *login* terlebih dahulu, sebagai lapisan keamanan dan personalisasi dalam sistem. Fitur mencari destinasi berkaitan dengan proses menambahkan destinasi ke perjalanan. Fitur untuk mengunggah gambar *cover* dan menambahkan catatan destinasi memperkaya detail rencana perjalanan, memungkinkan pengguna untuk memperkaya informasi perjalanan mereka.

## 2. Activity Diagram

### a. Autentikasi Akun



Gambar 3. 4 Activity Diagram Autentikasi Akun

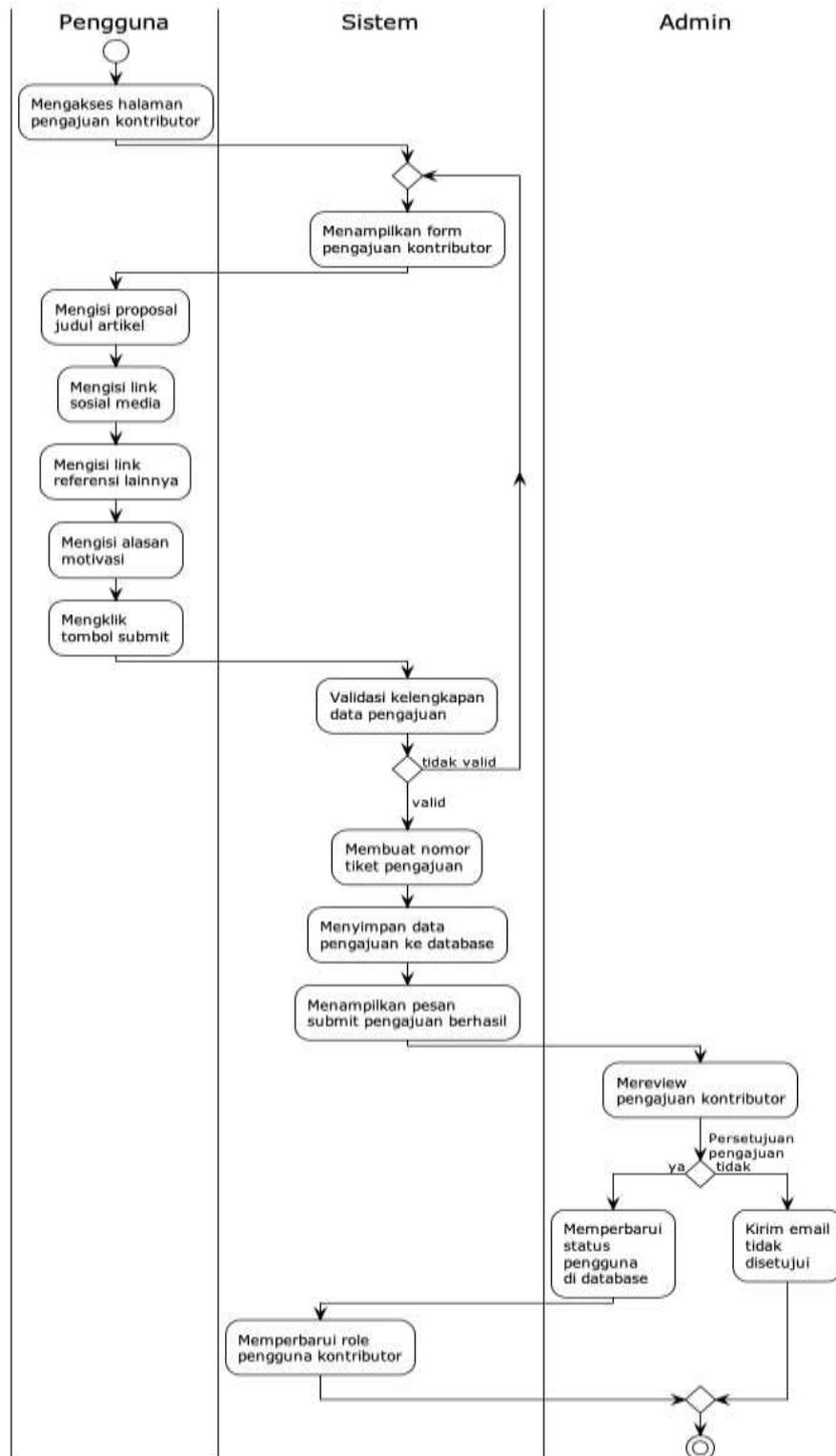
Pada diagram gambar 3.4 di atas menggambarkan alur interaksi antara pengguna dan sistem dalam proses *login* dan registrasi. Diagram ini dimulai dengan pengguna mengakses halaman *login*, dimana sistem akan menampilkan *form login*.

Pengguna yang sudah memiliki akun dapat langsung mengisi *form login* dengan memasukkan kredensial mereka (biasanya berupa email/username dan kata sandi). Setelah mengisi *form*, pengguna melakukan *submit* data. Sistem kemudian memproses input ini dengan melakukan validasi terhadap kredensial yang dimasukkan, membandingkannya dengan data yang tersimpan di *database*. Jika kredensial valid, sistem akan membuat sesi baru untuk pengguna, menyimpan informasi sesi ke dalam *cookie browser* untuk menjaga status *login*, dan akhirnya mengarahkan pengguna ke halaman beranda aplikasi. Proses ini memungkinkan pengguna untuk tetap terautentikasi selama mereka menggunakan aplikasi, tanpa perlu *login* ulang di setiap halaman.

Untuk pengguna yang belum memiliki akun, terdapat alur alternatif melalui proses registrasi. Pengguna dapat mengklik link registrasi yang tersedia di halaman *login*, yang kemudian mengarahkan mereka ke *form* registrasi. Pada *form* ini, pengguna diminta untuk mengisi data diri yang diperlukan untuk membuat akun baru, seperti nama, email, dan kata sandi. Setelah pengguna mengisi *form* dan melakukan *submit*, sistem melakukan serangkaian validasi terhadap data yang dimasukkan. Validasi ini mencakup pengecekan format email, panjang kata sandi, dan memastikan bahwa email belum terdaftar sebelumnya. Jika semua data valid, sistem akan menyimpan informasi pengguna baru ke dalam *database*. Selanjutnya, sama dengan proses *login*, sistem membuat sesi baru untuk pengguna yang baru terdaftar, menyimpan sesi ke dalam *cookie browser*, dan mengarahkan pengguna ke halaman beranda. Proses ini memastikan bahwa pengguna baru dapat langsung mengakses aplikasi setelah berhasil mendaftar, tanpa perlu melakukan *login*

tambahan. Penggunaan *cookie* untuk menyimpan sesi meningkatkan *user experience* dengan memungkinkan pengguna tetap terautentikasi selama sesi aktif, sehingga meningkatkan kenyamanan dan efisiensi penggunaan aplikasi.

## b. Pengajuan sebagai Kontributor/Blogger



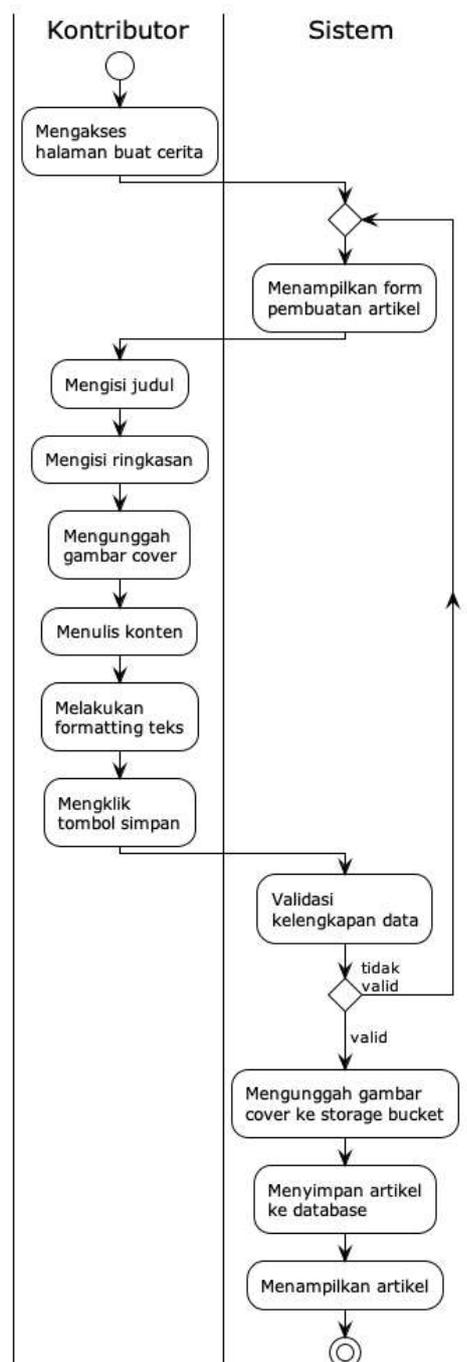
Gambar 3. 5 Activity Diagram Pengajuan Kontributor

Pada pengajuan kontributor melibatkan tiga aktor utama: pengguna, sistem, dan admin. Proses ini dimulai ketika pengguna mengakses halaman khusus pengajuan kontributor pada aplikasi. Sistem akan menampilkan *form* pengajuan kontributor, meminta pengguna untuk mengisi beberapa informasi. Informasi yang diminta mencakup proposal judul artikel yang ingin ditulis, tautan ke profil media sosial pengguna, link referensi lainnya yang dapat mendukung pengajuan, serta alasan dan motivasi pengguna untuk menjadi kontributor. Setelah mengisi semua informasi yang diperlukan, pengguna mengirimkan pengajuan dengan mengklik tombol *submit*. Sistem kemudian melakukan serangkaian validasi untuk memastikan kelengkapan dan kesesuaian data yang dimasukkan. Jika ditemukan ketidaklengkapan atau ketidaksesuaian, sistem akan mengarahkan pengguna kembali ke *form* untuk melakukan perbaikan atau melengkapi data yang kurang.

Setelah data pengajuan dinyatakan valid, sistem secara otomatis membuat nomor tiket unik untuk pengajuan tersebut. Data pengajuan lengkap beserta nomor tiket kemudian disimpan ke dalam *database* sistem. Sebagai konfirmasi, sistem menampilkan pesan sukses kepada pengguna, mengindikasikan bahwa pengajuan telah berhasil dikirim dan akan diproses lebih lanjut. Tahap berikutnya melibatkan peran admin yang akan mereview setiap pengajuan kontributor. Admin memiliki wewenang untuk menyetujui atau menolak pengajuan berdasarkan kriteria tertentu. Jika pengajuan disetujui, admin akan memperbarui status pengguna di *database* menjadi kontributor dan secara otomatis memberikan akses sebagai kontributor. Namun, jika pengajuan ditolak, admin akan mengirimkan email pemberitahuan kepada pengguna untuk menginformasikan bahwa pengajuannya tidak disetujui,

disertai dengan alasan penolakan jika ada. Proses ini dirancang secara menyeluruh untuk menjaga kualitas konten pada *platform* dengan memastikan bahwa hanya pengguna yang memenuhi standar dan kriteria tertentu yang dapat menjadi kontributor.

## c. Penulisan Artikel



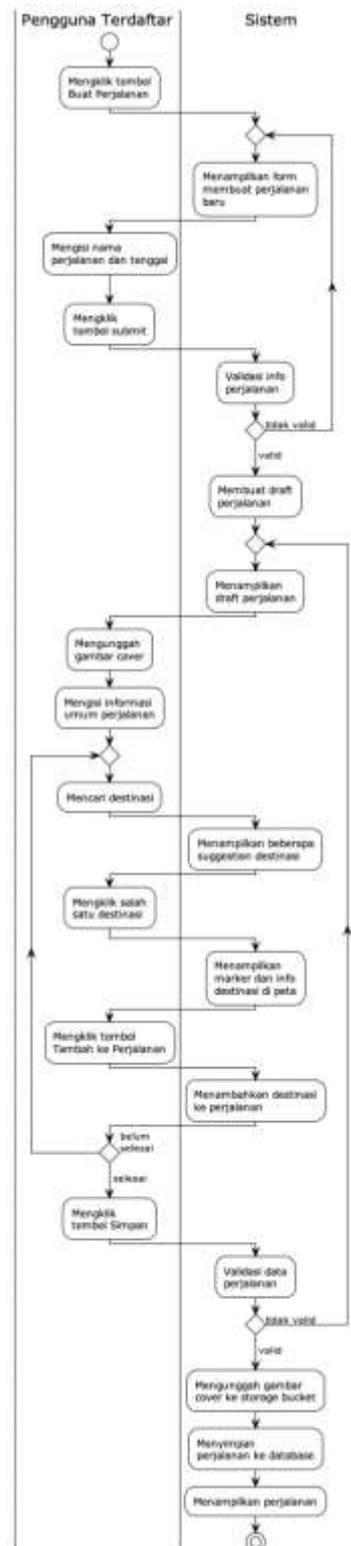
Gambar 3. 6 Activity Diagram Penulisan Artikel

*Activity diagram* di gambar 3.6, menggambarkan proses untuk kontributor membuat artikel cerita pada aplikasi. Proses dimulai ketika kontributor mengakses

halaman pembuatan cerita, dimana sistem menampilkan *form* pembuatan artikel. Kontributor kemudian melengkapi berbagai elemen artikel secara berurutan: mengisi judul yang menarik, menulis ringkasan yang menggambarkan isi artikel, mengunggah gambar *cover* yang relevan, menulis konten utama artikel, dan melakukan *formatting* teks untuk meningkatkan keterbacaan dan daya tarik visual. Setiap langkah ini untuk memudahkan kontributor dalam mengekspresikan cerita mereka secara efektif.

Setelah kontributor menyelesaikan *input* dan merasa puas dengan hasilnya, mereka dapat mengklik tombol simpan. Sistem akan melakukan validasi terhadap kelengkapan dan kesesuaian data, memastikan semua elemen wajib telah diisi dengan benar. Jika ditemukan ketidaklengkapan atau kesalahan, sistem akan berhenti dan meminta pengguna merevisi datanya. Setelah validasi berhasil, sistem melanjutkan dengan mengunggah gambar *cover* ke *storage bucket*, untuk dapat diakses publik. Seluruh data artikel, termasuk konten teks dan tautan gambar *cover*, kemudian disimpan ke dalam *database*. Lalu terakhir sistem menampilkan artikel yang baru saja dibuat, memberikan konfirmasi visual kepada kontributor bentuk artikel cerita yang sudah dipublikasi.

## d. Penyusunan Rencana Perjalanan



Gambar 3. 7 Activity Diagram Penyusunan Perjalanan

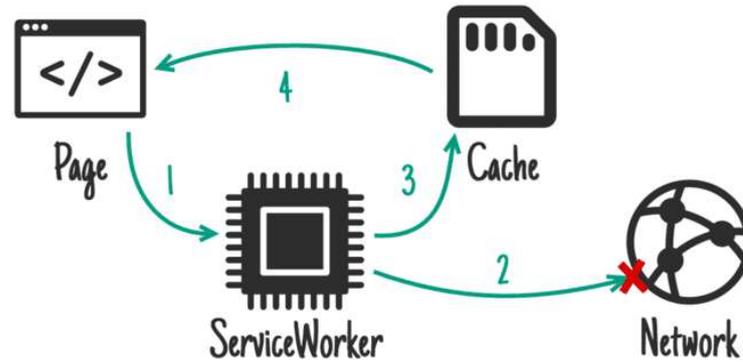
*Activity diagram* pembuatan rencana perjalanan dimulai ketika pengguna mengklik tombol "Buat Perjalanan", yang memicu sistem untuk menampilkan *form* pembuatan perjalanan baru. Pengguna kemudian mengisi informasi dasar seperti nama perjalanan dan tanggal, lalu mengirim *form* tersebut. Sistem melakukan validasi terhadap informasi yang dimasukkan, dan jika valid, sistem akan membuat *draft* perjalanan yang ditampilkan kepada pengguna.

Selanjutnya, proses menjadi lebih detail dan interaktif. Pengguna dapat mengunggah gambar *cover* perjalanan dan mengisi informasi umum perjalanan yang lebih lengkap. Fitur pencarian destinasi memungkinkan pengguna untuk mencari dan memilih destinasi spesifik, dengan sistem menampilkan *suggestion* dan informasi detail destinasi pada peta. Pengguna dapat menambahkan destinasi pilihan ke dalam rencana perjalanan mereka. Proses ini dapat diulang untuk menambahkan beberapa destinasi. Setelah pengguna merasa rencana perjalanan sudah lengkap, mereka dapat menyimpan perjalanan tersebut. Sistem kemudian melakukan validasi akhir, mengunggah gambar *cover* ke *storage bucket*, menyimpan seluruh data perjalanan ke *database*, dan akhirnya menampilkan rencana perjalanan yang telah dibuat. Alur ini mencerminkan implementasi yang memungkinkan pengguna untuk membuat rencana perjalanan yang terperinci dan terorganisir dengan mudah.

### 3. Kemampuan *Offline*

*Service worker* berfungsi sebagai pengelola event *fetch* yang terjadi di peramban, memainkan peran kunci dalam mengontrol bagaimana permintaan data diproses. Ia dapat menentukan apakah permintaan harus diteruskan ke server atau diambil dari *cache*, tergantung pada kondisi jaringan, apakah dalam keadaan daring atau luring. Diantara berbagai strategi *caching* yang tersedia untuk *service worker*, dua strategi utama yang akan digunakan adalah *network first* dan *cache first*. Dalam strategi ini, *service worker* pertama-tama memeriksa apakah jaringan memberikan respon yang valid. Jika koneksi berhasil dan data tersedia, maka informasi tersebut dikembalikan langsung ke halaman web. Namun, jika koneksi gagal, *service worker* akan mengambil data dari *cache* yang sudah ada. Strategi ini sangat berguna untuk situasi di mana diperlukan data yang selalu diperbarui, seperti respon dari API, tetapi juga penting untuk memastikan tampilan konten yang berguna ketika jaringan tidak dapat diakses.

Ketika ada koneksi internet, *service worker* harus meneruskan setiap permintaan dari halaman web ke server. Setelah menerima tanggapan dari server, *service worker* akan menyimpan salinan tanggapan tersebut dalam *cache browser*. Kemudian tanggapan dari server dikirimkan kembali ke halaman web yang diminta. Dengan cara ini, pengguna secara otomatis dapat mengakses versi *offline cache* dari setiap halaman yang mereka kunjungi. Meskipun pengguna dapat mengakses konten tersebut secara *offline*, mereka tidak akan dapat mengirimkan data baru ke server selama tidak ada koneksi internet.



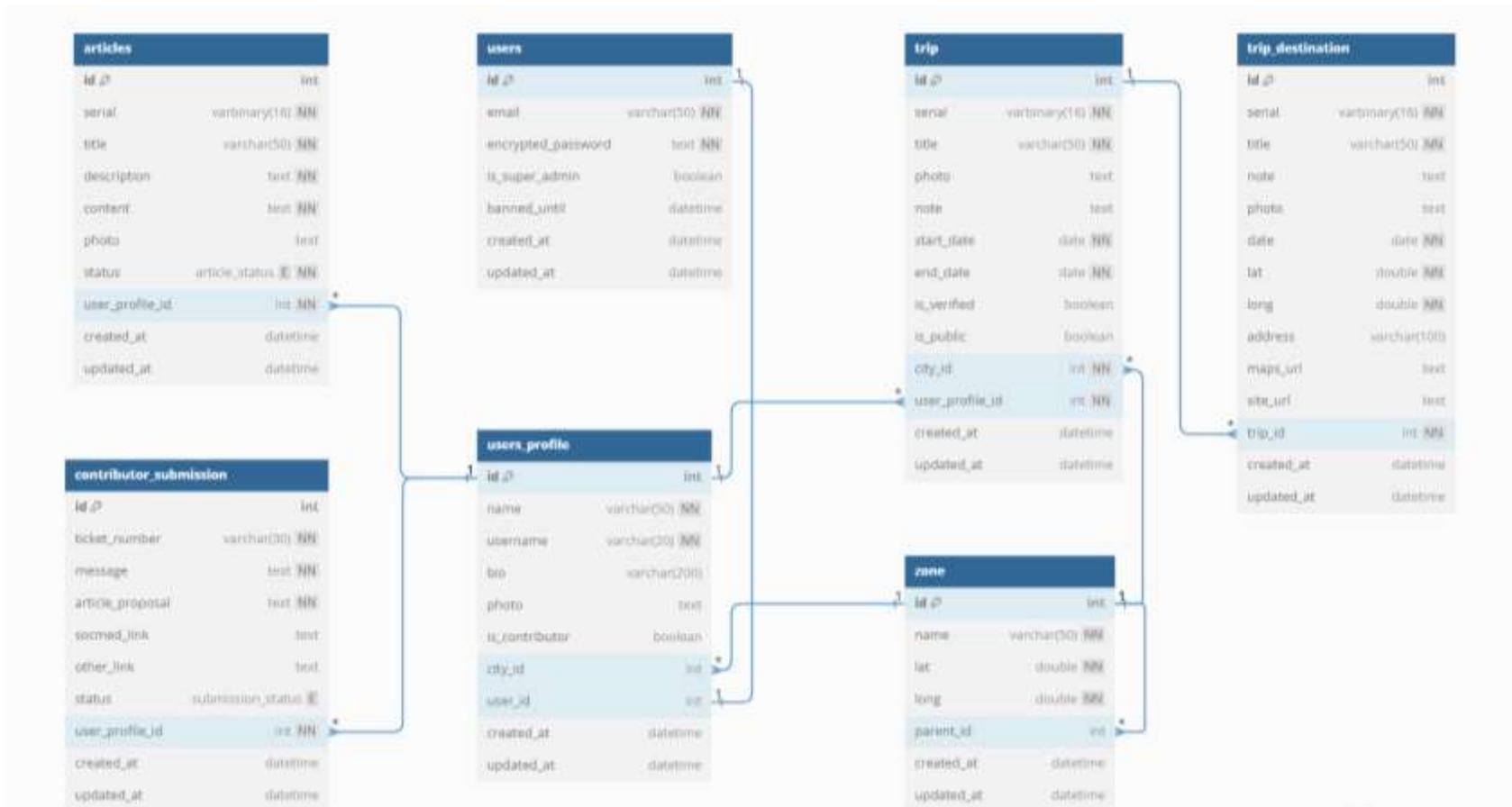
Gambar 3. 8 Service worker dalam keadaan offline

Halaman akan dikembalikan dari cache ketika kondisi *offline* atau tidak ada respon dari server. Hal ini terjadi jika halaman sudah ada di dalam cache sebelumnya, seperti yang ditunjukkan pada Gambar 3.8..

### 3.2.2 Perancangan Data

Dalam proses perancangan data untuk aplikasi ini dirancang untuk mengakomodasi berbagai fitur dan fungsi yang ditawarkan. Pertama, data dirancang untuk mengelola konten artikel cerita, yang termasuk teks, gambar, dan metadata terkait. Kedua, sistem data juga dioptimalkan untuk mendukung fitur pembuatan rencana perjalanan, memungkinkan pengguna untuk menyusun dan menyimpan itinerari mereka dengan mudah. Selain itu, perancangan data juga mempertimbangkan proses pengajuan kontributor, yang mungkin melibatkan penyimpanan informasi tentang calon kontributor, karya contoh mereka, dan status aplikasi. Terakhir, struktur data dirancang untuk menangani pengaturan akun pengguna dan sistem autentikasi yang aman, menjamin privasi dan keamanan informasi pribadi pengguna. Semua aspek perancangan data ini bertujuan untuk menciptakan pengalaman pengguna yang mulus dan efisien, sambil memastikan skalabilitas dan kinerja aplikasi yang optimal.

Perancangan data aplikasi pada gambar 3.9 meliputi beberapa tabel yang saling berelasi. Tabel 'users' berfungsi untuk menyimpan data autentikasi pengguna seperti email dan kata sandi, sementara tabel 'users\_profile' yang berelasi *one-to-one* dengan 'users', menyimpan informasi profil pengguna seperti nama, foto, dan biodata. Pemisahan kedua tabel ini meningkatkan keamanan dengan memungkinkan kontrol akses yang lebih ketat pada data sensitif, meningkatkan performa *query*, memberikan fleksibilitas dalam pengelolaan data profil, dan menyediakan skalabilitas untuk pengembangan di masa depan. Tabel 'trip' menyimpan informasi umum rencana perjalanan, sementara 'trip\_destination' menyimpan detail destinasi dalam setiap perjalanan. Tabel 'article' menampung konten artikel cerita dari kontributor, dan 'contributor\_submission' mencatat pengajuan untuk menjadi kontributor. Tabel 'zone' menyimpan informasi lokasi seperti kota dan provinsi, dengan relasi ke dirinya sendiri (*self-referencing*) untuk merepresentasikan hierarki antara kota dan provinsi. Struktur hierarki ini memungkinkan penyimpanan dan *query* yang efisien untuk data geografis yang berjenjang. Struktur data ini secara komprehensif mendukung fitur-fitur utama aplikasi, memungkinkan manajemen pengguna yang aman, perencanaan perjalanan dan berbagi cerita yang terperinci, serta pengelolaan data lokasi yang fleksibel.

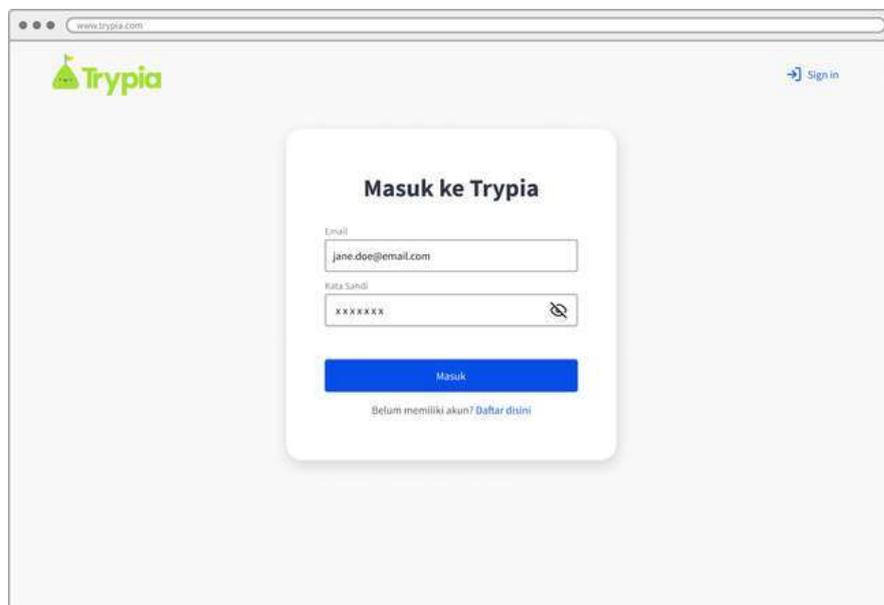


Gambar 3. 9 ERD Aplikasi

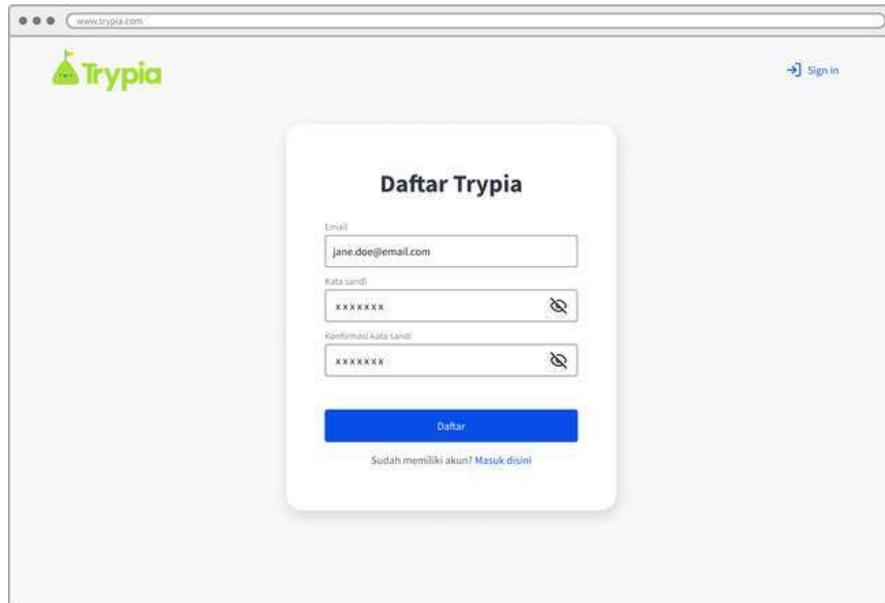
### 3.2.3 Perancangan *User Interface* / *Mock-up* aplikasi

Pada perancangan *user interface/mock-up* aplikasi, dijabarkan rancangan antarmuka aplikasi secara keseluruhan. *Mock-up* aplikasi menjadi acuan dasar mengimplementasikan *user interface* pada kode program. *Mock-up* dapat memvisualkan perancangan tata letak dan struktur setiap elemen antarmuka pengguna sebelum implementasi sebenarnya. Perancangan ini dapat memastikan bahwa desain aplikasi memenuhi kebutuhan pengguna dan memberikan pengalaman yang optimal. Terdapat beberapa mockup sederhana yang mewakili aktivitas pada *activity* diagram di sub-bab perancangan sistem.

Berikut adalah mock-up untuk halaman *login* dan *register* yang didesain untuk memberikan pengalaman autentikasi yang sederhana bagi pengguna. Tampilannya bersih dan terstruktur untuk memastikan pengguna mudah dalam memasukkan informasi yang diperlukan untuk mengakses fitur-fitur aplikasi.



Gambar 3. 10 Autentikasi Login

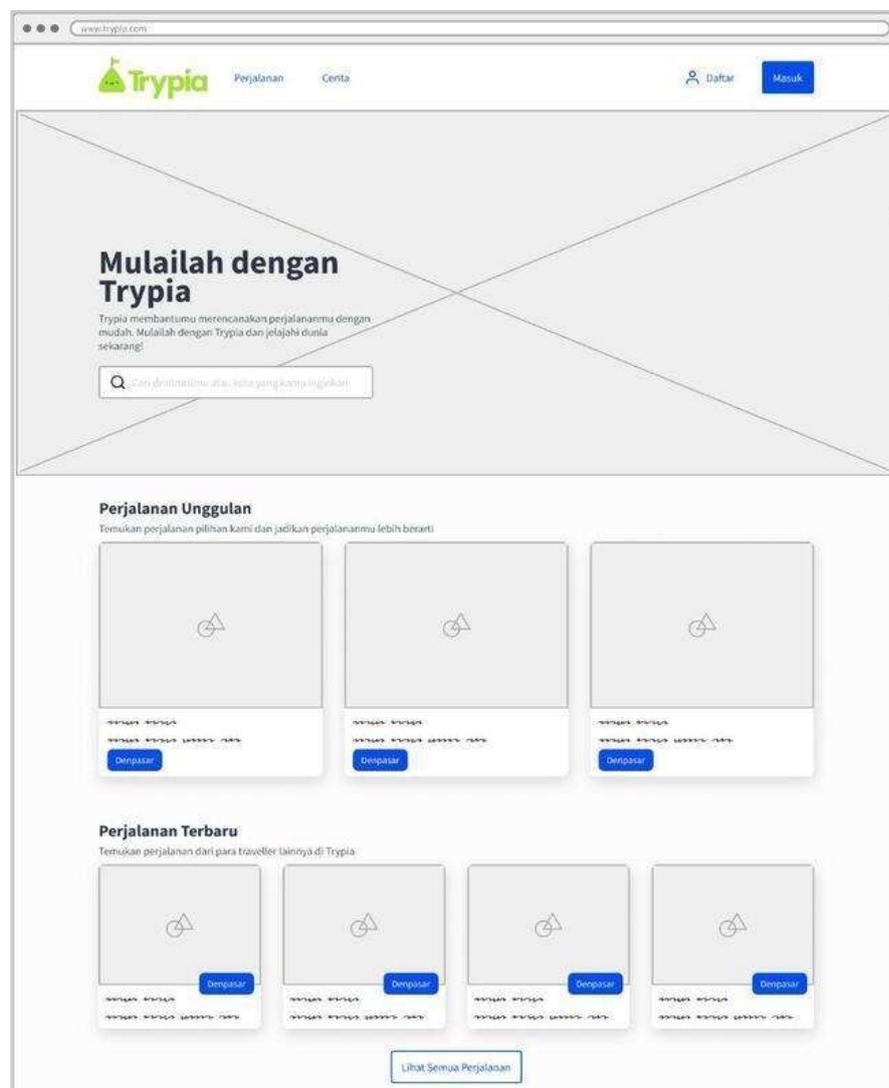


The image shows a web browser window with the URL `www.trypia.com`. The page features the Trypia logo in the top left and a "Sign in" link in the top right. The main content is a registration form titled "Daftar Trypia". The form includes the following fields and elements:

- Email:** A text input field containing the email address `jane.doe@email.com`.
- Kata sandi:** A password input field with masked characters `xxxxxxx` and a toggle icon to show/hide the password.
- Konfirmasi kata sandi:** A confirmation password input field with masked characters `xxxxxxx` and a toggle icon to show/hide the password.
- Daftar:** A prominent blue button for submitting the registration form.
- Footer:** A link that reads "Sudah memiliki akun? Masuk disini" (Already have an account? Log in here).

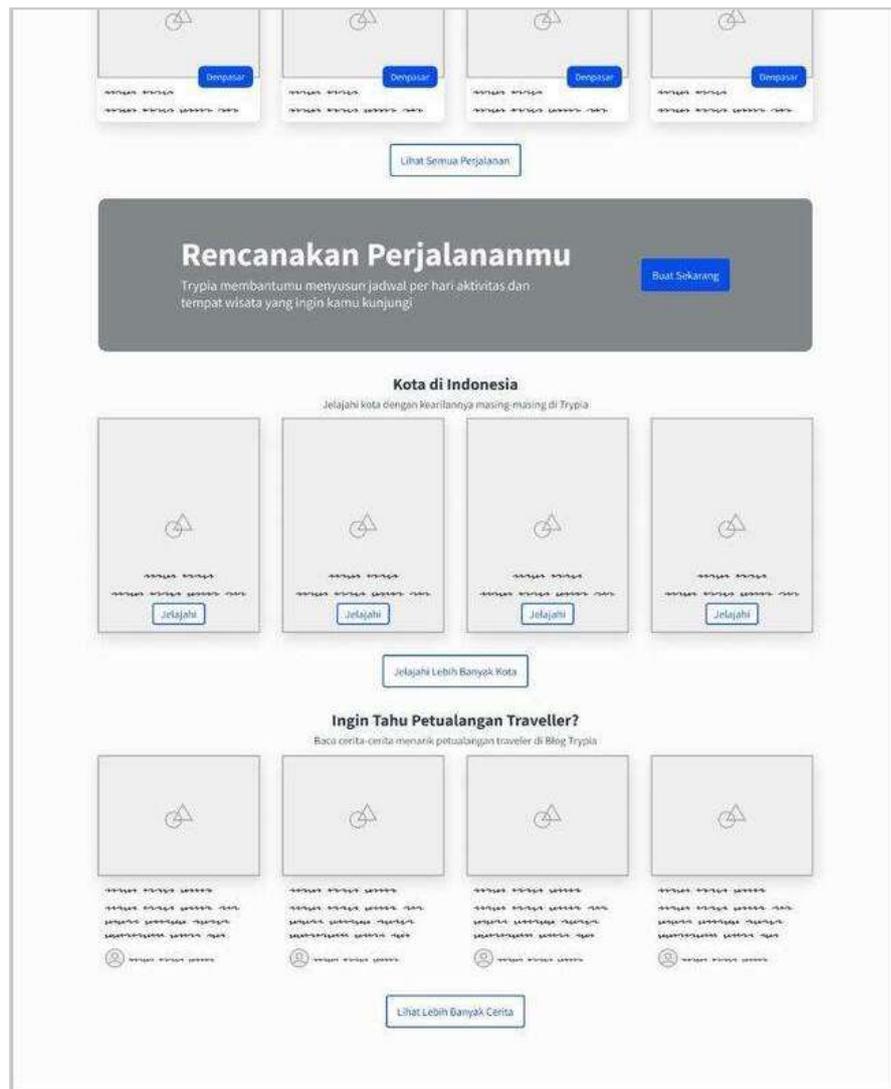
Gambar 3. 11 Autentikasi Register

Pada halaman beranda, terdapat *jumbotron* dengan kata-kata jargon untuk memberikan kesan pertama dari persona aplikasi. Lalu di bawahnya terdapat *list* rencana perjalanan yang dikelompokkan menjadi dua kategori yaitu, perjalanan unggulan dan perjalanan terbaru. Perjalanan unggulan adalah rencana perjalanan yang ditandai khusus oleh admin, sementara perjalanan terbaru adalah daftar perjalanan yang diurutkan berdasarkan tanggal dibuatnya rencana perjalanan tersebut.



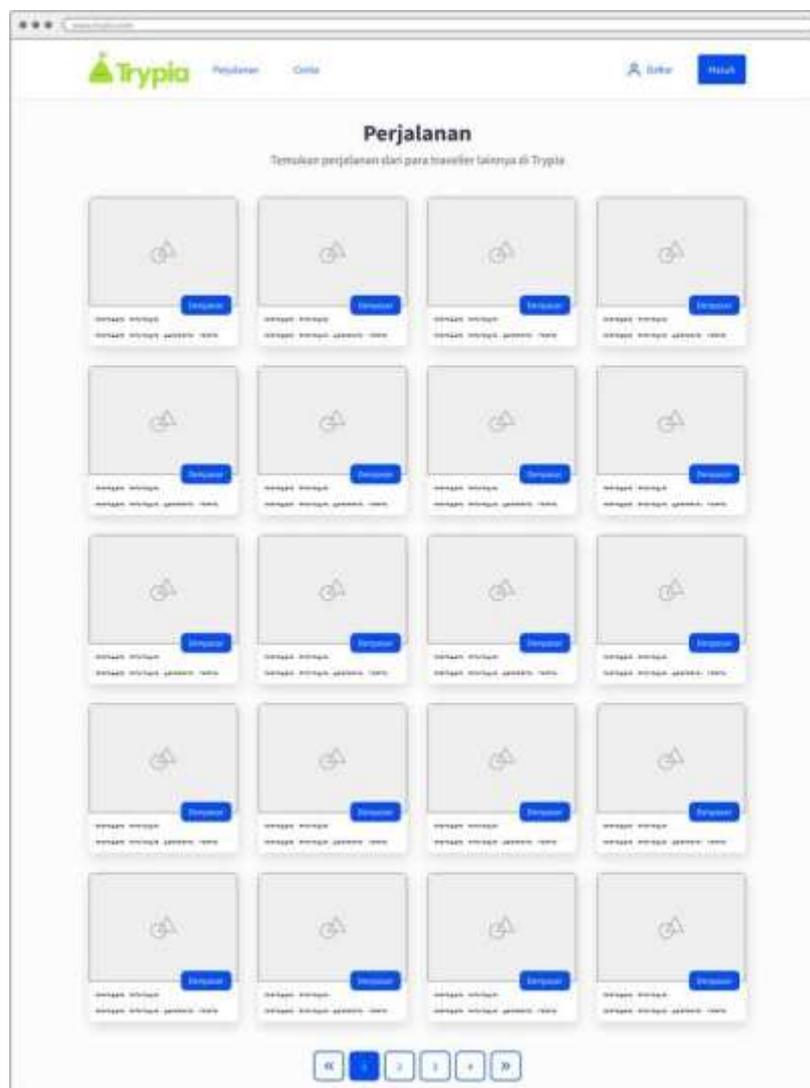
Gambar 3. 12 Beranda Bagian Atas

Selain itu dibagian bawah *list* perjalanan, terdapat *call to action* (CTA) untuk membuat perjalanan baru. Lalu di bawah CTA, pengguna dapat melihat beberapa kota yang tersedia data perjalanannya di aplikasi. Dilanjutkan dibagian bawahnya lagi menampilkan beberapa artikel cerita untuk memberikan pengguna inspirasi dan informasi menarik terkait kisah dari kreator lainnya.



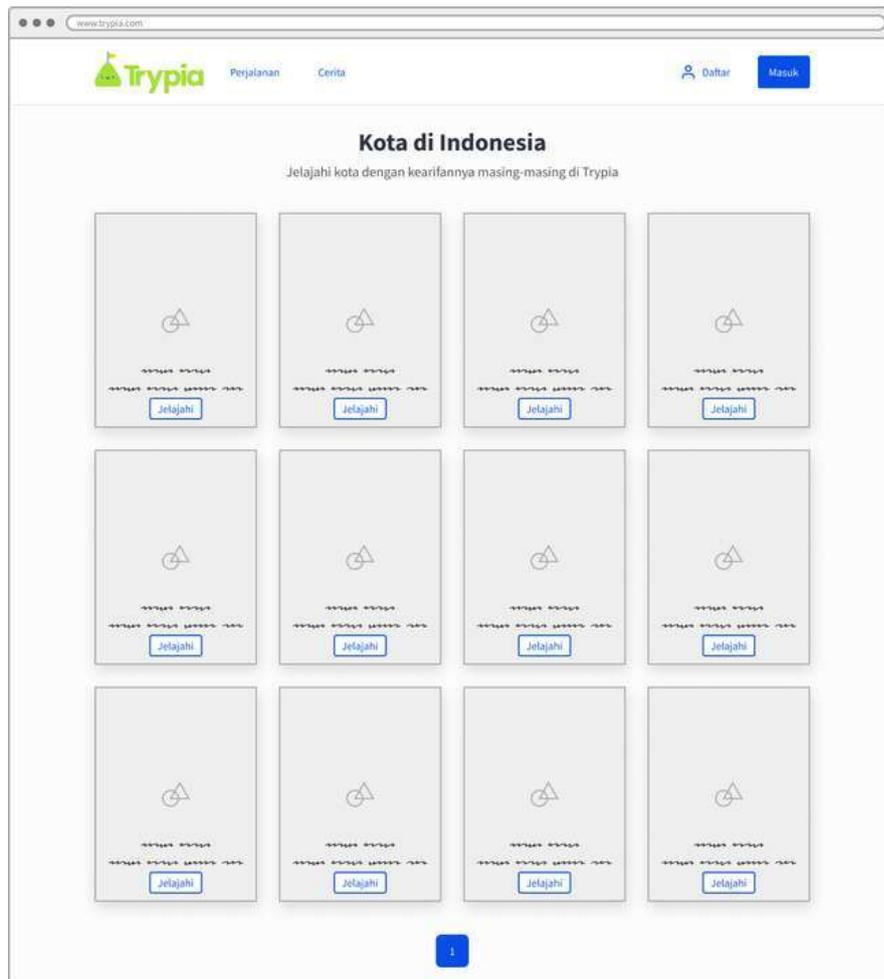
Gambar 3. 13 Beranda Bagian Bawah

Pada bagian *list* perjalanan terdapat tombol "Lihat semua perjalanan", saat diklik maka akan dialihkan ke halaman khusus *list* perjalanan. Halaman ini menampilkan beberapa rencana perjalanan dari kreator lain dengan urutan terbaru di atas, dan dibagi ke beberapa halaman dengan *pagination*. Setiap perjalanan di *list* akan menampilkan foto, judul perjalanan, dan kota tujuan, sehingga pengguna yang sedang mencari atau melihat-lihat rencana perjalanan dapat dengan mudah menemukan perjalanan yang menarik atau mereka butuhkan.



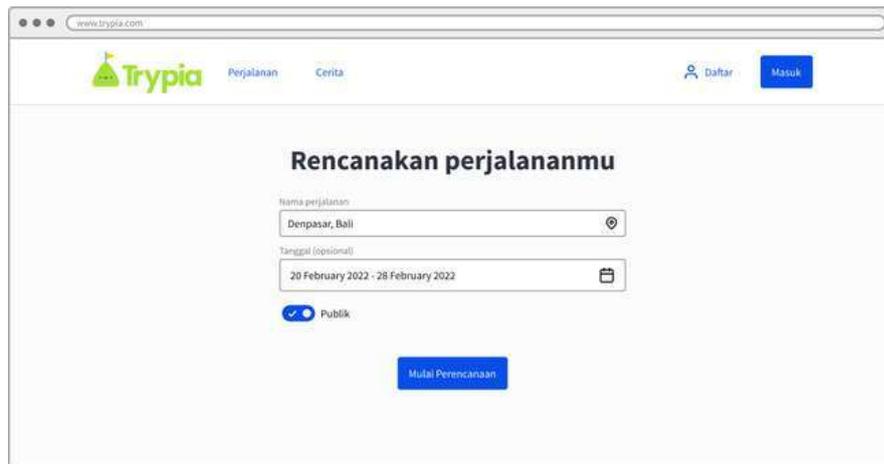
Gambar 3. 14 List Perjalanan

Selain *list* perjalanan umum, terdapat juga *list* perjalanan yang dikelompokkan berdasarkan kota. *List* ini dapat dibuka dengan mengklik salah satu kota yang ada dibagian kota pada halaman beranda. Saat pengguna memilih kota, maka akan dialihkan ke halaman yang menampilkan *list* perjalanan yang ada di kota tersebut, disertai dengan foto, judul perjalanan, dan informasi lainnya.



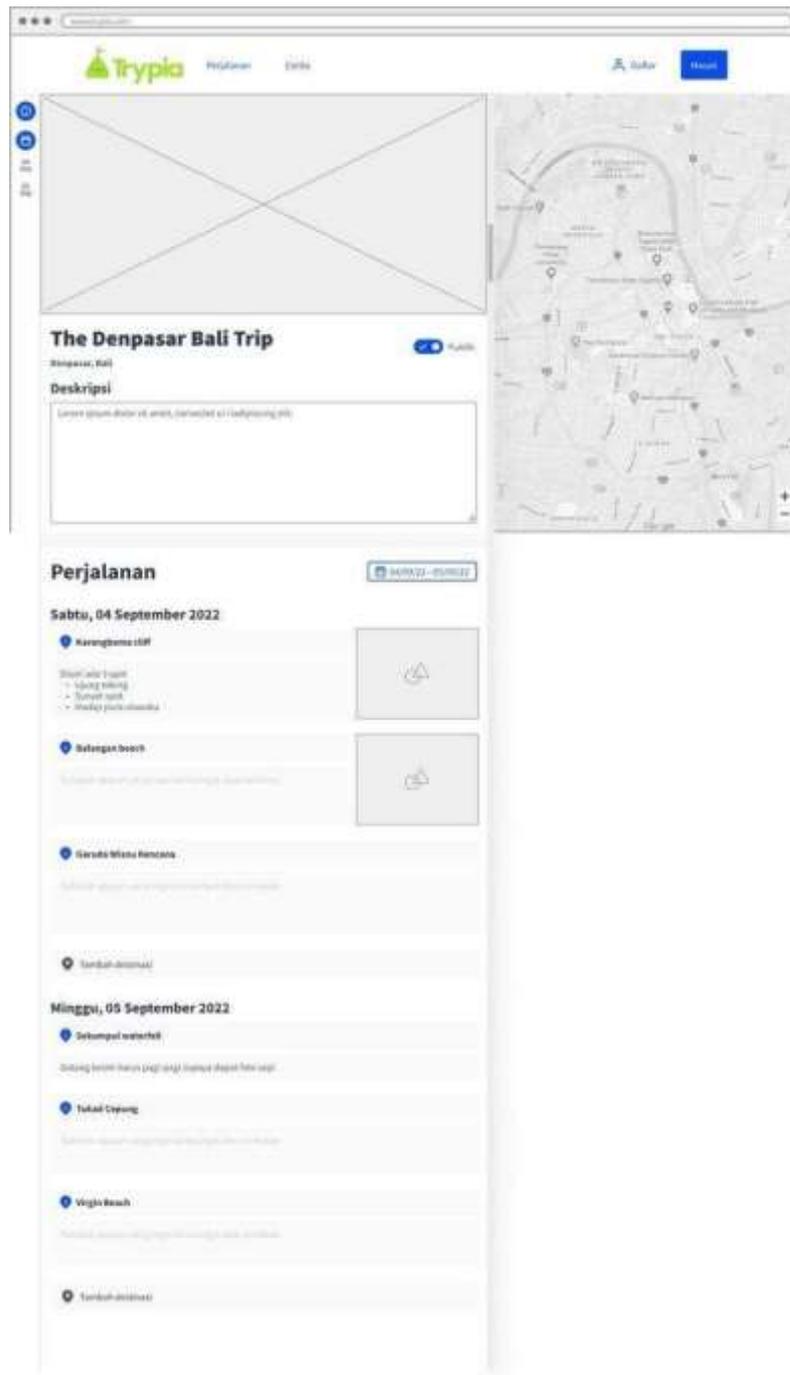
Gambar 3. 15 List Perjalanan berdasarkan Kota

Setiap perjalanan dibuat oleh pengguna melalui *form* membuat perjalanan. *Form* ini terdiri dari beberapa input, termasuk judul perjalanan, tanggal perjalanan, dan status perjalanan. Desain *form* terdiri dari 2 langkah, yang pada langkah pertama berupa *form* sederhana untuk memastikan pengguna tidak merasa rumit saat akan membuat perjalanan.

The image shows a web browser window with the URL www.trypia.com. The page features the Trypia logo and navigation links for 'Perjalanan' and 'Cerita'. There are 'Daftar' and 'Masuk' buttons in the top right. The main heading is 'Rencanakan perjalananmu'. Below it, there is a form with three input fields: 'Nama perjalanan' with the value 'Denpasar, Bali', 'Tanggal (opsional)' with the value '20 February 2022 - 28 February 2022', and a 'Publik' checkbox which is checked. A blue button labeled 'Mulai Perencanaan' is positioned below the form.

Gambar 3. 16 *Form* Membuat Perjalanan

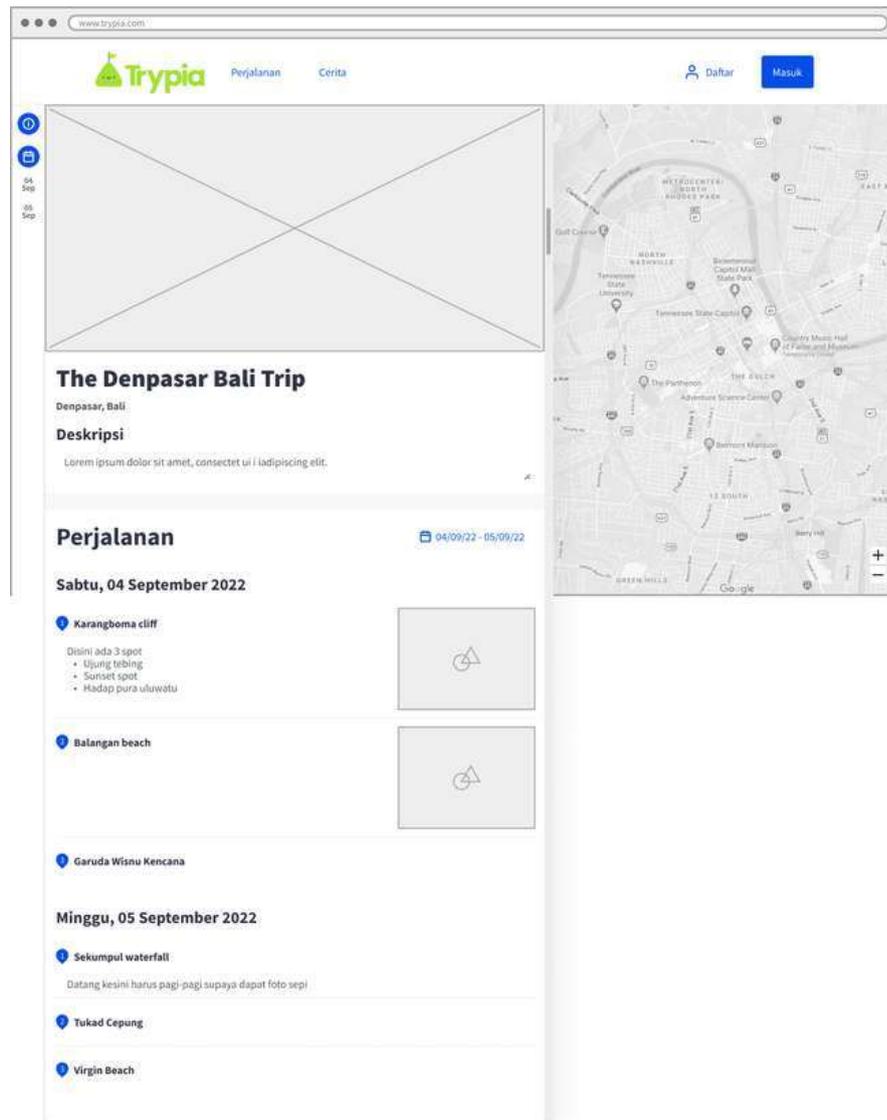
Setelah *submit form* langkah pertama, akan langsung dialihkan ke halaman langkah kedua, yaitu halaman detail membuat perjalanan. Di halaman ini pengguna dapat mengunggah gambar *cover*, mengganti judul, menambahkan kota, dan memberikan deskripsi perjalanan. Untuk destinasi dari perjalanan dapat ditambahkan dengan mencari nama destinasi di fitur *suggestion*, lalu lokasinya akan ditampilkan di peta lengkap beserta informasi lokasi tersebut. Selain itu, pengguna dapat menambahkan catatan pada setiap destinasi untuk memberikan informasi tambahan atau detail khusus.



Gambar 3. 17 Detail Membuat Perjalanan

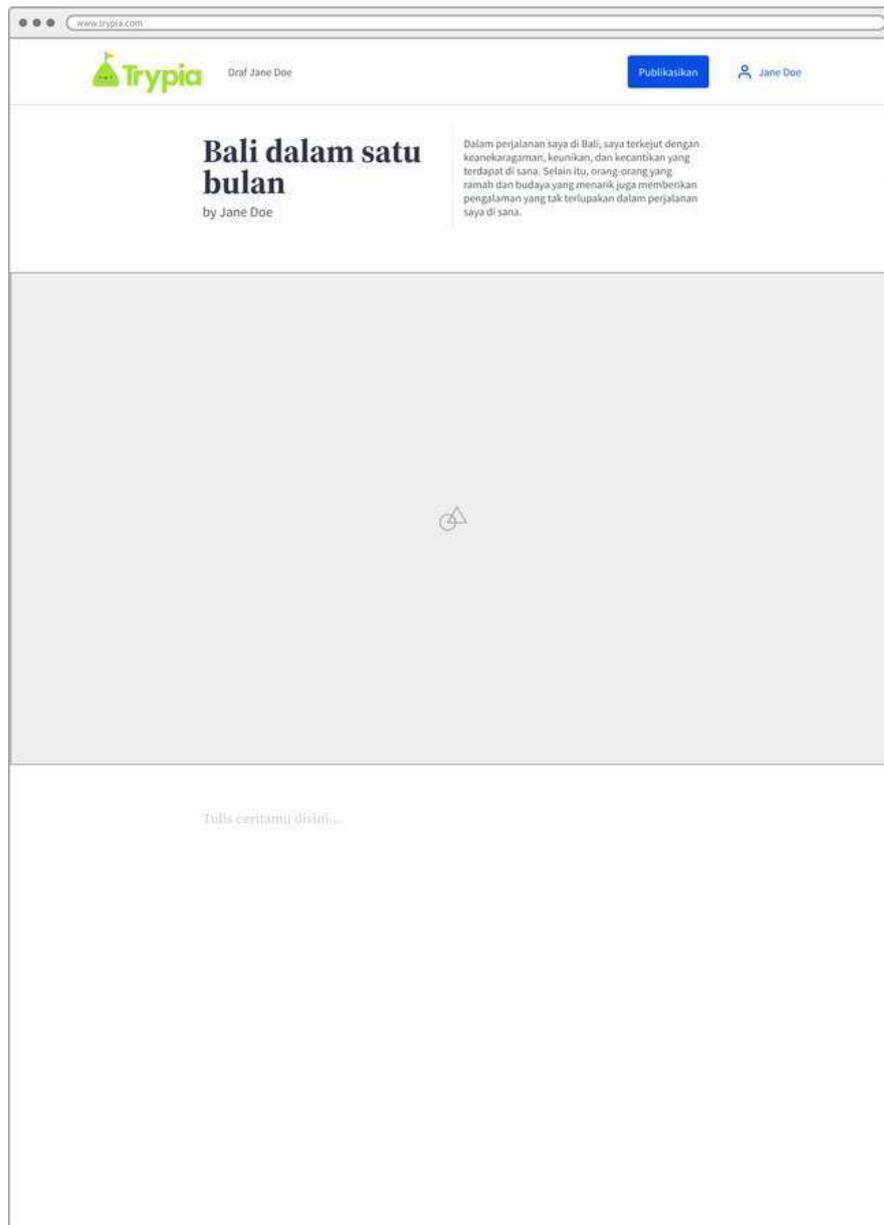
Rencana perjalanan yang sudah dibuat pengguna dapat dilihat oleh pengguna lain jika statusnya adalah publik. Halaman detail perjalanan ini dibuat agar pengguna dapat fokus melihat rencana perjalanan secara lengkap. Pengguna

dapat melihat gambar *cover*, judul perjalanan, kota yang akan dikunjungi, dan deskripsi perjalanan. Lalu terdapat juga peta yang menunjukkan lokasi destinasi yang termasuk dalam rencana perjalanan. Setiap destinasi dilengkapi dengan catatan yang memberikan detail tambahan atau informasi penting terkait destinasi tersebut.



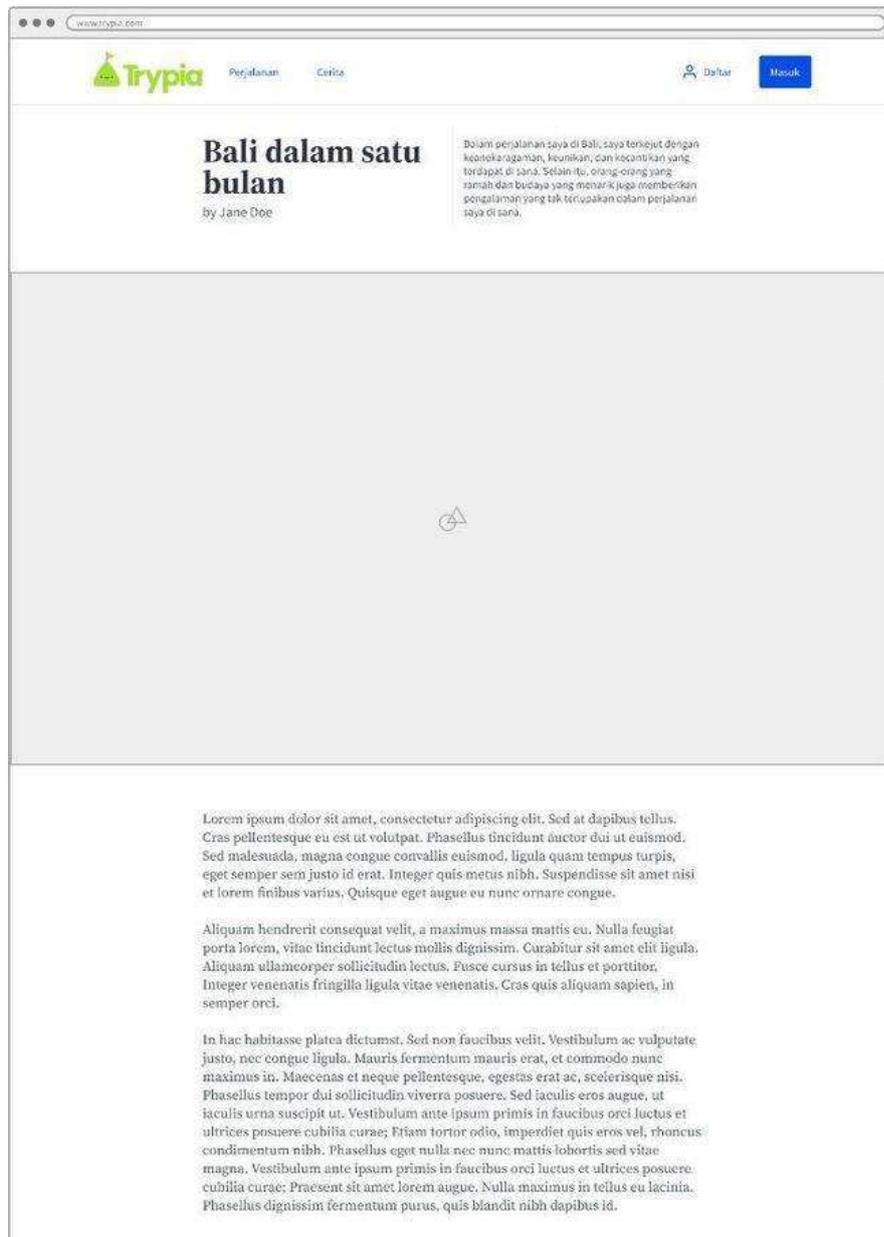
Gambar 3. 18 Detail Rencana Perjalanan

Ada juga artikel cerita, disini pengguna dapat membuat artikel untuk menceritakan kisah perjalanannya. Pengguna dapat mengisi gambar artikel, judul, ringkasan cerita, dan konten artikel itu sendiri. Desain *form* menulis artikel sama dengan mode *view*-nya, sehingga saat menulis pengguna dapat langsung tergambarakan akan tampil seperti apa artikel ceritanya nanti. Tata letak artikel cerita cukup sederhana untuk mempermudah pengguna dalam menulis dan membaca. Di bawah gambar *cover* merupakan editor teks dengan berbagai *formatting text* untuk memberikan keleluasaan bagi pengguna berkreasi menuliskan ceritanya.



Gambar 3. 19 Membuat Artikel Cerita

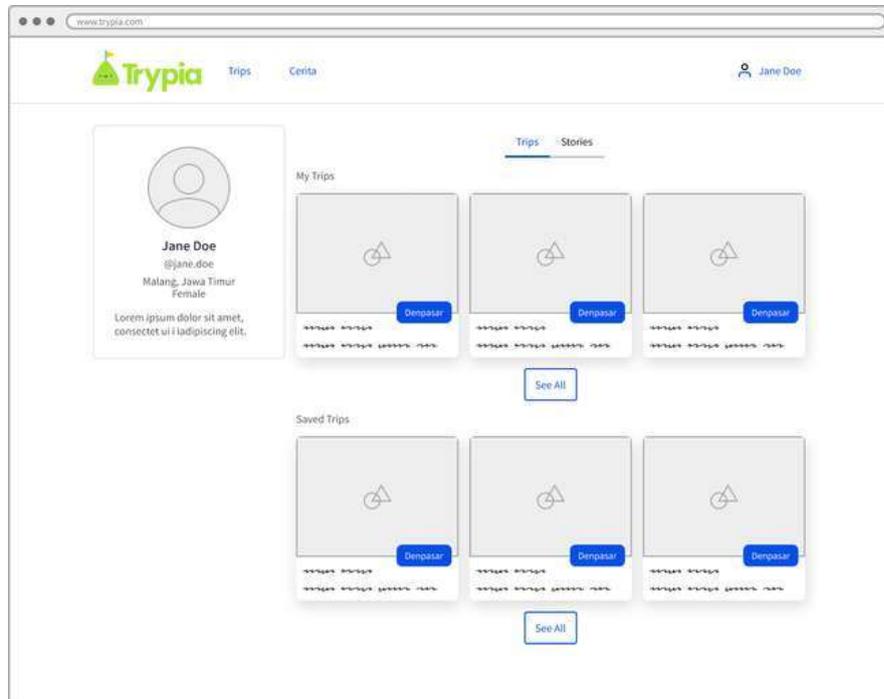
Pada halaman detail artikel cerita mode *view* menampilkan artikel cerita secara lengkap dengan gambar, judul, ringkasan, dan kontennya. Desain halaman ini dibuat agar pengguna untuk membaca artikel dengan nyaman, menikmati visual yang sederhana agar dapat fokus dan mendapatkan informasi mendalam dari konten yang dibaca.



Gambar 3. 20 Detail Artikel Cerita

Lalu ada halaman profil pengguna yang menampilkan data diri pengguna, seperti foto profil, nama, *username*, kota domisili, dan bio singkat. Di sebelahnya terdapat *list* perjalanan dan artikel cerita yang pernah dibuat oleh pengguna. Setiap perjalanan dan artikel cerita ditampilkan seperti pada halaman beranda. Di halaman

profil ini, pengguna dapat melihat kontribusi dan aktivitas pengguna yang mungkin menarik baginya.



Gambar 3. 21 Profil Pengguna

Profil pengguna dapat diubah melalui pengaturan akun. Pengguna dapat memperbarui detail pribadi seperti nama, kota, bio, dan foto profil. Selain itu ada juga halaman pengaturan kata sandi untuk pengguna mengganti kata sandi akun aplikasinya. Pada halaman ini, pengguna diminta untuk memasukkan kata sandi baru mereka, kata sandi lama mereka, dan kode konfirmasi untuk masing-masing.

The screenshot shows a web browser window with the URL 'www.trypia.com'. The page header includes the 'Trypia' logo, navigation links for 'Trips' and 'Cerita', and a user profile icon for 'Jane Doe'. The main content area is titled 'Settings' and has two sub-sections: 'Account' (selected) and 'Security'. Under 'Account', there is a profile picture placeholder and several input fields: 'Username' (jane.doe), 'Name' (Jane Doe), 'Email' (jane.doe@email.com), 'City' (a dropdown menu showing 'Malang'), and 'Bio' (a text area with placeholder text). At the bottom, there are radio buttons for 'Gender' (Male, Female, Prefer not disclose) and an 'Update' button.

Gambar 3. 22 Pengaturan Profil

The screenshot shows the same web browser window as above, but the 'Security' sub-section is selected. It contains three password input fields: 'Current password', 'New password', and 'Confirm new password'. Each field has a toggle icon to show or hide the password. The 'Confirm new password' field has a small note below it: 'Confirm new password (8 to 32 characters long)'. A 'Change password' button is located at the bottom of the form.

Gambar 3. 23 Pengaturan Ganti Password

### 3.3 Rancangan Pengujian

Pengujian sistem dilakukan dengan metode pengujian langsung; pendekatan Black Box Testing hanya menguji fungsi dan output sistem tanpa memperhatikan struktur internal atau kode sumbernya. Metode ini dipilih karena fokus pada pengujian fungsi aplikasi tanpa perlu mengetahui struktur internal atau kode

sumbernya. Oleh karena itu, pengujian *black box* memungkinkan peneliti untuk mengumpulkan serangkaian kondisi *input* yang memenuhi semua persyaratan fungsional program. *Black box testing* dilakukan dengan 2 tahap pengujian, yaitu *Functional Testing* dan *Acceptance Testing*.

### 1. *Functional Testing*

*Functional testing* berkaitan dengan pengujian keseluruhan fungsi sistem. Pengujian ini akan dilakukan untuk menentukan terpenuhinya fungsi aplikasi sesuai dengan spesifikasi teknis. Yang diujikan adalah fitur-fitur aplikasi dan penerapan teknologi PWA dan TWA. Selain itu, dilakukan juga pengujian terhadap implementasi *Trusted Web Activities* (TWA), yaitu mengenai pengujian aplikasi yang diinstall melalui file apk. Pengujian ini ditujukan untuk membuktikan bahwa aplikasi web dapat diinstall sebagai aplikasi android, yang berarti aplikasi tersebut juga bisa didistribusikan melalui *app market store* seperti *Google Play Store*.

### 2. *Acceptance Testing*

Pengujian ini berkaitan dengan pemenuhan sistem terhadap kriteria kebutuhan dari pengguna, persyaratan, dan proses bisnis yang dilakukan. Pada pengujian ini akan dilakukan kepada orang yang tidak memiliki kaitan dengan proses pengembangan maupun proses pengujian sebelumnya dan terbiasa melakukan kegiatan wisata, secara spesifik yaitu anggota dari komunitas Belantara.co. Lingkup yang diujikan mirip dengan spesifikasi dari *functional testing* namun fokusnya lebih memastikan bahwa aplikasi memenuhi kebutuhan bisnis dan siap diterima oleh pengguna akhir yaitu wisatawan.