

BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Saat magang di PT Truclimate Dekarbonisasi Indonesia, mahasiswa bertugas sebagai Mobile Junior Fullstack Developer. Mahasiswa merancang dan mengembangkan aplikasi DMRV untuk klien perusahaan. Peserta magang menerima bimbingan langsung dari supervisornya yaitu, Bapak Jeffrey Dharmawan, sebagai Chief Technology Officer (CTO) pada PT TruClimate Dekarbonisasi Indonesia. Peserta magang berinteraksi dengan berbagai tim selama proses koordinasi magang. Tim-tim ini termasuk tim carbon, developer, dan designer. Peserta magang pada kesempatan ini akan melakukan kerja magang hybrid WFO dan WFH. WFO dilakukan pada hari Senin, Rabu, dan Jumat, dan WFH dilakukan Selasa dan Kamis. Berikut ini adalah pertemuan antara peserta magang dan supervisor.



Gambar 3. 1 Tampilan Meeting dengan Tim TruClimate

Untuk masalah komunikasi atau koordinasi proyek di PT TruClimate Dekarbonisasi Indonesia, WhatsApp digunakan sebagai platform utama, baik melalui grup maupun pesan pribadi untuk berinteraksi. Alternatif metode komunikasi lainnya dapat dilihat pada Tabel 3.1.

Tabel 3. 1 Koordinasi dan Komunikasi Tim Software Developer

Kegiatan	Frekuensi Kegiatan	Tujuan
Daily Update	Setiap hari (selama project berlangsung)	Memberikan update mengenai hal yang sudah dilakukan (baik itu pembuatan fitur / perbaikan design)
Weekly Meeting	Setiap sebulan sekali	<ol style="list-style-type: none"> 1. Membahas update personal terkait perubahan yang dilakukan kepada supervisor 2. Supervisi mengkoordinasi untuk melakukan perubahan terhadap suatu hal

3.1.1 Weekly Meeting

Weekly Meeting biasanya diadakan setiap hari Senin, dengan frekuensi satu kali per minggu. Pertemuan ini membahas pekerjaan yang telah dilakukan selama periode satu minggu, misalnya dari Senin hingga Senin berikutnya. Dalam *Weekly Meeting* ini, setiap staf dan peserta magang akan mempresentasikan hasil pekerjaan mereka kepada seluruh tim dan CEO untuk berbagi capaian selama satu minggu tersebut.

3.1.2 Sharing Session

Sharing session merupakan pertemuan yang ditujukan bagi tim karbon untuk melakukan presentasi atau memberikan penjelasan terkait wawasan tentang proyek karbon. Proyek karbon ini nantinya akan diimplementasikan sesuai dengan kebutuhan masing-masing tim.

3.2 Tugas dan Uraian Kerja Magang

Sebagai peserta magang dengan posisi *Mobile Junior Developer* di PT TruClimate Dekarbonisasi Indonesia, tugas utamanya adalah mengembangkan aplikasi mobile bernama DMRV. Aplikasi ini digunakan dalam konteks proyek karbon dan lingkungan untuk mengelola data terkait pengukuran, pelaporan, dan verifikasi emisi gas rumah kaca atau aktivitas lingkungan lainnya. DMRV

membantu memantau, merekam, dan melaporkan data guna memastikan bahwa proyek berjalan sesuai dengan standar yang ditetapkan, seperti pengurangan emisi karbon atau peningkatan keberlanjutan.

Aplikasi ini memungkinkan pengguna mengumpulkan data secara digital, sehingga proses verifikasi dan pelaporan menjadi lebih efisien dan akurat, serta memfasilitasi audit oleh pihak ketiga yang memverifikasi klaim lingkungan atau karbon dari sebuah proyek. Proyek ini akan berlangsung dari tanggal 26 Juni 2024 hingga 25 Oktober 2024. Tanggung jawab peserta magang mencakup pengembangan aplikasi hingga tahap *deployment*, serta penyusunan laporan magang. Pembagian tugas secara lebih rinci dapat dilihat pada Tabel 3.2 di bawah ini.

Tabel 3. 2 Waktu Pelaksanaan Magang Perusahaan

No	Uraian Pekerjaan	Minggu	Tanggal Mulai	Tanggal Selesai
1.	Pembuatan konsep design dan antarmuka	1	26 Juni 2024	28 Juni 2024
	Melakukan penginstalan software yang dibutuhkan untuk kebutuhan project	1	28 Juni 2024	30 Juni 2024
	Melakukan eksekusi project dengan cara mendesign UI di figma	1	28 Juni 2024	22 Juli 2024
	Mencari referensi template atau desain yang tepat untuk memastikan aplikasi DMRV tampil menarik, fungsional, dan mudah digunakan.	1	28 Juni 2024	2 Juli 2024
	Membuat UI halaman onboarding	2	2 Juli	3 Juli

Membuat UI halaman Authentication (Login, Forget Password, OTP) beserta case yang terjadi	2	4 Juli	5 Juli
Membuat UI component page dari halaman yang akan digunakan nantinya	2	8 Juli	11 Juli
Menentukan icon apa saja yang akan digunakan untuk asset aplikasi	2	11 Juli	12 Juli
Melakukan design pada halaman dashboard (Task Progress, Your Tasks, Project Overview)	2	15 Juli	19 Juli
Membuat Project overview screen dan List project	2	22 Juli	25 Juli
Membuat Settings screen, change password, dan project history	2	26 Juli	31 Juli
Membuat semua filter modal effect pada button filter yang telah di apply	3	1 Agustus	2 Agustus
Membuat project details, add record beserta component didalamnya	4	5 Agustus	7 Agustus
Melakukan evaluasi terhadap design yang	4	7 Agustus	9 Agustus

	telah dibuat terkait bagian apa saja yang perlu diperbaiki			
2.	Menghasilkan aplikasi mobile yang sepenuhnya berfungsi dan memenuhi standar yang ditentukan oleh Perusahaan	5	12 Agustus	13 Agustus
	Instalasi package dependencies, membuat semua widget yang akan digunakan dan layout terkait design yang sudah direncanakan di figma	5	13 Agustus	6 September
3.	Membantu kerjasama tim teknis dalam perencanaan dan pengembangan aplikasi mobile	6	7 September	7 Oktober
	Membuat endpoint pada masing masing rest api yang akan digunakan untuk kebutuhan <i>frontend</i>	9	9 September	4 Oktober
4.	Melakukan testing / pengujian secara manual terhadap aplikasi mobile	10	8 Oktober	26 Oktober
	Melakukan testing per masing-masing page dan fitur dari aplikasi mobile yang telah dibuat	10	8 Oktober	26 Oktober

3.2.1 Penjelasan Mengenai Software dan Framework yang digunakan

Berdasarkan Tabel 3.2, tabel ini menjelaskan rincian jadwal dan tahapan pekerjaan dalam pelaksanaan magang yang bertujuan untuk mengembangkan aplikasi mobile, dimulai dari persiapan desain hingga pengujian aplikasi. Tahap pertama dimulai pada minggu pertama hingga minggu keempat, di mana fokus awal adalah membuat konsep desain dan antarmuka aplikasi. Pada minggu pertama, dilakukan pembuatan konsep desain, pemasangan software yang diperlukan untuk proyek, mencari referensi desain untuk memastikan aplikasi memiliki tampilan yang menarik dan fungsional, serta memulai desain UI di Figma.

Pada minggu kedua, pengerjaan berlanjut ke pembuatan UI untuk halaman onboarding dan autentikasi, serta komponen yang diperlukan pada halaman-halaman aplikasi. Ikon-ikon yang akan digunakan juga ditentukan pada minggu ini, dan desain halaman dashboard mulai dikerjakan. Pada minggu ketiga, dilakukan pembuatan filter modal dan layar project overview. Minggu keempat dikhususkan untuk evaluasi desain, guna memperbaiki bagian yang dirasa masih kurang sesuai atau perlu penyesuaian.

Pada tahap pengembangan aplikasi yang berlangsung di minggu kelima dan keenam, dilakukan pembuatan aplikasi mobile agar berfungsi penuh dan memenuhi standar perusahaan. Tahap ini mencakup instalasi paket dependencies serta pembuatan layout dan widget berdasarkan desain yang sudah direncanakan di Figma. Di minggu keenam, tim berkolaborasi dalam perencanaan serta pengembangan aplikasi mobile agar mencapai hasil yang optimal.

Di minggu kesembilan, dilakukan pembuatan endpoint untuk REST API yang nantinya akan digunakan frontend, di mana ini merupakan bagian dari implementasi backend. Tahap akhir adalah pengujian aplikasi yang berlangsung pada minggu kesepuluh, di mana dilakukan pengujian manual terhadap aplikasi mobile, mencakup setiap halaman dan fitur yang ada untuk

memastikan aplikasi berfungsi sesuai harapan. Tabel ini memberikan gambaran terstruktur mengenai timeline dan tugas-tugas yang dilaksanakan, dari tahap awal desain hingga pengujian akhir dalam pengembangan aplikasi mobile selama program magang.

Setelah menyelesaikan pembuatan *prototype* yang dikerjakan selama satu bulan *sebelumnya*, langkah selanjutnya adalah mempersiapkan berbagai tools dan *software* yang akan mendukung proses *development*. Persiapan ini sangat penting untuk memastikan bahwa seluruh tim pengembang, baik di sisi *frontend* maupun *backend*, memiliki lingkungan kerja yang optimal. *Tools* ini mencakup perangkat lunak desain, platform pengembangan, serta alat bantu untuk mengelola kode dan database.

3.2.1.1 Visual Studio Code

Visual Studio Code adalah sebuah aplikasi IDE (*Integrated Development Environment*) yang populer dan ringan, digunakan oleh banyak pengembang untuk menulis dan mengelola kode dalam berbagai bahasa pemrograman. IDE ini dipilih karena memiliki dukungan yang luas untuk berbagai *extension*, yang dapat memperluas fungsionalitasnya sesuai dengan kebutuhan proyek.



Gambar 3. 2 Visual Studio Code Logo

3.2.1.2 Docker

Docker adalah sebuah platform yang digunakan untuk mengotomatiskan proses penyebaran aplikasi dalam container, yaitu unit yang mengemas

aplikasi beserta seluruh dependensi dan lingkungan yang dibutuhkan untuk berjalan. Dengan Docker, pengembang dapat memastikan bahwa aplikasi mereka berjalan konsisten di berbagai lingkungan, mulai dari komputer pengembang hingga server produksi, tanpa perlu khawatir tentang masalah kompatibilitas atau konfigurasi.



Gambar 3. 3 Docker

3.2.1.3 DBeaver

DBeaver adalah sebuah database management tool yang digunakan untuk mengelola dan berinteraksi dengan berbagai jenis database secara grafis. Alat ini mendukung beragam database management systems (DBMS) seperti MySQL, PostgreSQL, SQLite, Oracle, SQL Server, dan banyak lainnya, sehingga memudahkan pengguna untuk bekerja dengan berbagai sistem database dari satu antarmuka yang sama.



Gambar 3. 4 DBeaver

3.2.1.4 Postman

Postman adalah sebuah alat yang digunakan untuk menguji dan mengelola API (Application Programming Interface), yang sangat membantu pengembang dalam membangun dan mengintegrasikan layanan berbasis web. Dengan Postman, pengguna dapat membuat dan mengirim berbagai jenis HTTP requests seperti GET, POST, PUT, dan DELETE, serta melihat hasil respons yang diberikan oleh server, termasuk kode status, body respons, dan waktu yang dibutuhkan untuk respons tersebut.



POSTMAN

Gambar 3. 5 Postman

Salah satu fitur utama Postman adalah kemampuannya untuk menyimpan dan mengatur koleksi request. Hal ini memungkinkan pengguna untuk menjalankan pengujian secara berulang tanpa harus menulis ulang request dari awal setiap kali melakukan pengujian. Postman juga menawarkan dukungan untuk berbagai jenis autentikasi seperti OAuth, API Key, atau JWT, sehingga mempermudah proses pengujian API yang memerlukan otentikasi.

3.2.1.5 Flutter

Flutter adalah sebuah framework open-source yang dikembangkan oleh Google untuk membangun aplikasi nativel yang bisa berjalan di berbagai platform, seperti Android, iOS, web, dan desktop, menggunakan satu basis kode yang sama. Dengan menggunakan bahasa pemrograman Dart, Flutter memungkinkan pengembang untuk membuat antarmuka pengguna yang responsif dan menarik dengan performa tinggi.



Gambar 3. 6 Framework Flutter

Salah satu keunggulan Flutter adalah konsep widget-nya, di mana hampir segala sesuatu dalam Flutter, mulai dari layout hingga elemen UI, dianggap sebagai widget. Hal ini memungkinkan pengembang untuk dengan mudah menggabungkan dan menyesuaikan berbagai elemen untuk membuat antarmuka pengguna yang dinamis. Widget ini bersifat deklaratif, sehingga

ketika ada perubahan pada data, tampilan dapat diperbarui dengan cepat dan efisien.

3.2.1.6 Node js

Node.js adalah sebuah runtime environment berbasis JavaScript yang digunakan untuk menjalankan kode JavaScript di luar browser, terutama di sisi server. Dikembangkan oleh Ryan Dahl pada tahun 2009, Node.js memungkinkan pengembang untuk menggunakan JavaScript tidak hanya untuk pengembangan *frontend*, tetapi juga untuk *backend*, sehingga menciptakan keseragaman bahasa pemrograman di seluruh stack pengembangan.



Gambar 3. 7 Framework Node Js

Salah satu kekuatan utama Node.js adalah kemampuannya untuk menangani I/O yang bersifat non-blocking, yang berarti bahwa Node.js sangat efisien dalam menangani banyak permintaan sekaligus tanpa harus menunggu setiap operasi selesai. Ini sangat cocok untuk aplikasi yang memerlukan real-time processing, seperti aplikasi chat, streaming, atau API yang membutuhkan banyak request. Model event-driven ini memungkinkan Node.js untuk menangani skala besar pengguna dengan konsumsi sumber daya yang lebih rendah dibandingkan dengan server tradisional yang bersifat blocking.

3.2.1.7 MySql

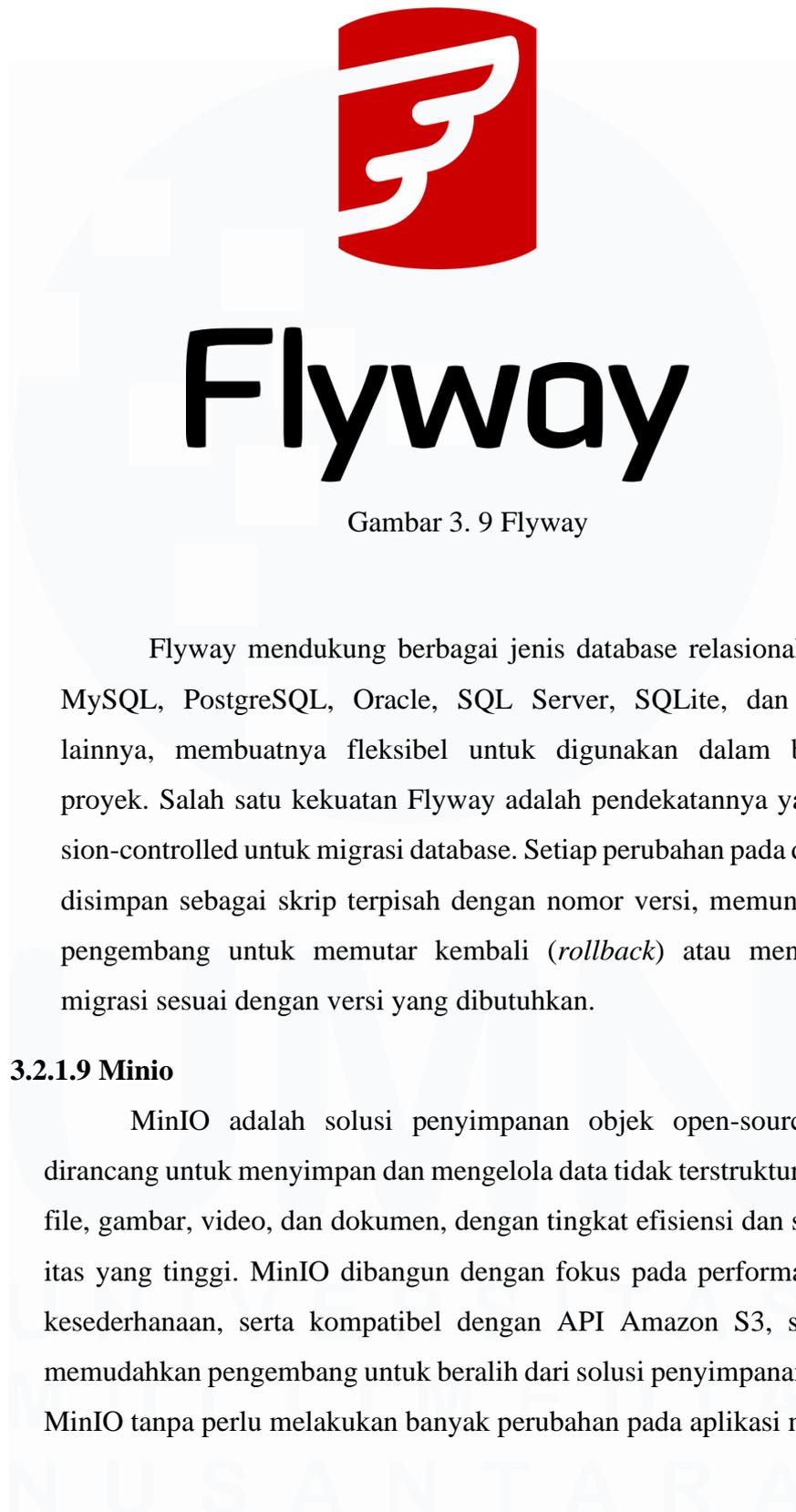
MySQL adalah sistem manajemen basis data relasional (RDBMS) *open-source* yang banyak digunakan untuk menyimpan, mengelola, dan mengambil data. MySQL menggunakan bahasa SQL (*Structured Query Language*) sebagai bahasa standar untuk mengelola data di dalam database-nya. Dikembangkan pertama kali oleh MySQL AB dan sekarang dimiliki oleh Oracle Corporation, MySQL dikenal karena kinerjanya yang cepat, skalabilitasnya yang tinggi, dan fleksibilitasnya, sehingga menjadi pilihan populer di kalangan pengembang dan perusahaan untuk berbagai jenis aplikasi.



Gambar 3. 8 MySQL

3.2.1.8 Flyway

Flyway adalah alat *database migration* yang digunakan untuk mengelola perubahan pada skema basis data secara otomatis dan terstruktur. Ini membantu pengembang dalam melacak dan menerapkan perubahan pada struktur database, seperti penambahan tabel baru, modifikasi kolom, atau perubahan lainnya, secara konsisten di seluruh lingkungan pengembangan, pengujian, dan produksi. Flyway bekerja dengan mengatur dan menjalankan skrip SQL atau Java yang berisi migrasi skema secara berurutan.



Gambar 3. 9 Flyway

Flyway mendukung berbagai jenis database relasional seperti MySQL, PostgreSQL, Oracle, SQL Server, SQLite, dan banyak lainnya, membuatnya fleksibel untuk digunakan dalam berbagai proyek. Salah satu kekuatan Flyway adalah pendekatannya yang version-controlled untuk migrasi database. Setiap perubahan pada database disimpan sebagai skrip terpisah dengan nomor versi, memungkinkan pengembang untuk memutar kembali (*rollback*) atau menerapkan migrasi sesuai dengan versi yang dibutuhkan.

3.2.1.9 Minio

MinIO adalah solusi penyimpanan objek open-source yang dirancang untuk menyimpan dan mengelola data tidak terstruktur, seperti file, gambar, video, dan dokumen, dengan tingkat efisiensi dan skalabilitas yang tinggi. MinIO dibangun dengan fokus pada performansi dan kesederhanaan, serta kompatibel dengan API Amazon S3, sehingga memudahkan pengembang untuk beralih dari solusi penyimpanan lain ke MinIO tanpa perlu melakukan banyak perubahan pada aplikasi mereka.



Gambar 3. 10 Minio

Salah satu fitur unggulan MinIO adalah kemampuannya untuk menyimpan data dalam mode *distributed*, yang memungkinkan pengguna untuk mengelola dan menyimpan data di beberapa node, menjadikannya cocok untuk penyimpanan berskala besar. MinIO mendukung *erasure coding*, yang meningkatkan ketersediaan dan keandalan data dengan membagi data menjadi beberapa bagian dan menyimpannya di beberapa lokasi, sehingga data tetap dapat diakses meskipun terjadi kegagalan pada salah satu node.

3.2.2 Perancangan Proses Bisnis

3.2.2.1 Proses Bisnis Berjalan

Proses bisnis sebelum adanya aplikasi DMRV masih dilakukan secara manual, di mana pencatatan data dilakukan menggunakan kertas. Setiap data yang dikumpulkan di lapangan kemudian dicatat secara manual oleh petugas, baik itu data terkait proyek, informasi plot, maupun perkembangan tugas yang telah diselesaikan. Namun, metode manual ini menimbulkan berbagai masalah yang mempengaruhi efisiensi dan akurasi data.

Beberapa masalah yang muncul dari sistem manual ini antara lain:

1. Ketidaktepatan Data: karena pencatatan dilakukan secara manual, terdapat risiko tinggi terjadinya kesalahan dalam input data,

seperti salah tulis, hilangnya informasi penting, atau tidak konsistennya data yang dicatat dari lapangan.

2. Keterlambatan Sinkronisasi Data: Dalam sistem manual, data yang dikumpulkan di lapangan tidak dapat langsung diakses oleh pihak lain secara real-time. Hal ini mengakibatkan keterlambatan dalam penyampaian informasi yang penting, terutama jika data baru diinput dan diserahkan dalam waktu yang lama setelah kegiatan di lapangan selesai.
3. Kesulitan dalam Memantau Perkembangan Proyek: Dengan proses pencatatan manual, sangat sulit untuk memantau perkembangan proyek secara keseluruhan. Data proyek yang tersebar dalam bentuk dokumen fisik menyebabkan kesulitan dalam melakukan pelacakan progres secara akurat dan terintegrasi.

Akibat dari permasalahan-permasalahan tersebut, manajemen proyek menjadi kurang efisien, baik dalam hal pengambilan keputusan, pelaporan, maupun pengawasan.

3.2.2.2 Proses Bisnis yang Diusulkan

Sebagai solusi atas berbagai masalah yang timbul dari sistem manual, aplikasi DMRV diusulkan untuk mendigitalisasi proses bisnis yang berjalan. Aplikasi ini dirancang untuk mempermudah pengelolaan data proyek dan plot secara real-time dan terintegrasi, sehingga dapat memberikan efisiensi lebih tinggi dalam pengelolaan proyek lapangan.

Beberapa fitur utama yang diusulkan melalui aplikasi DMRV ini meliputi:

1. Sinkronisasi Data Otomatis dan Manual: Aplikasi DMRV memungkinkan pengguna untuk melakukan sinkronisasi data

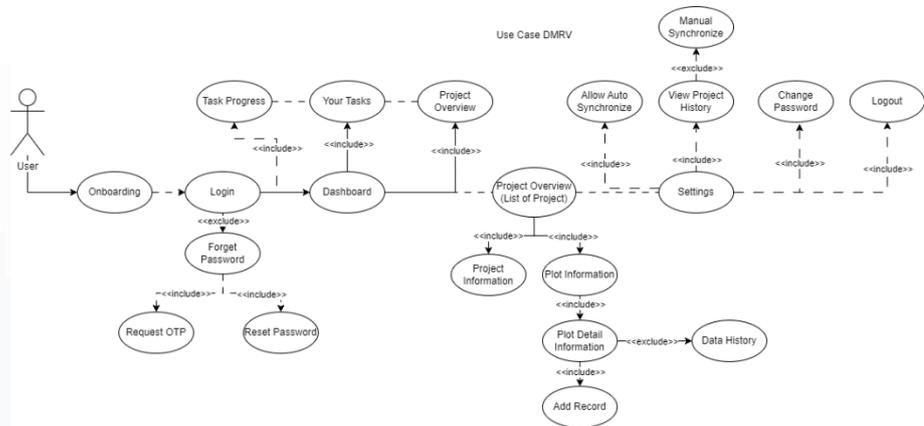
lapangan secara otomatis atau manual dengan server. Hal ini akan mengurangi risiko kesalahan dalam pencatatan data serta meminimalkan keterlambatan dalam pengumpulan data dari lapangan.

2. **Pembaruan Data Secara Real-time:** Pengguna dapat memperbarui dan mengakses informasi proyek serta perkembangan tugas secara langsung melalui aplikasi. Dengan begitu, setiap perubahan yang terjadi di lapangan dapat langsung terupdate pada sistem, memastikan data yang diterima oleh tim selalu up-to-date.
3. **Pengelolaan Tugas yang Lebih Efektif:** Melalui dasbor aplikasi, pengguna dapat dengan mudah melihat tugas-tugas mereka, memantau perkembangan tugas yang sudah dan sedang dilakukan, serta memperoleh gambaran yang jelas tentang status keseluruhan proyek. Hal ini memungkinkan pengelolaan tugas yang lebih terstruktur dan efisien.

Dengan adanya sistem ini, permasalahan seperti keterlambatan sinkronisasi, kesalahan pencatatan, dan kesulitan memantau perkembangan proyek dapat diminimalisir. Sistem yang terintegrasi ini mendukung efisiensi operasional serta peningkatan akurasi dan kecepatan dalam manajemen proyek.

3.2.2.3 Design System

Berikut merupakan gambar use case diagram aplikasi DMRV yang terdapat pada gambar 3.11



Gambar 3.11 Use Case Diagram DMRV

Use Case Diagram untuk aplikasi DMRV menggambarkan alur interaksi antara pengguna dengan berbagai fitur utama dalam sistem. Diagram ini menyoroti beberapa skenario penting yang terkait dengan pengelolaan data dan sinkronisasi informasi dalam proyek lingkungan.

Pada awalnya, pengguna akan melalui proses Onboarding untuk memahami fungsi aplikasi. Setelah itu, mereka dapat Login ke dalam sistem untuk mengakses dashboard, yang berfungsi sebagai pusat kendali. Jika pengguna mengalami masalah saat masuk, seperti lupa kata sandi, ada pilihan untuk menggunakan fitur Forget Password yang mencakup permintaan OTP dan pengaturan ulang kata sandi.

Setelah berhasil masuk, pengguna akan diarahkan ke Dashboard, di mana mereka dapat memantau Task Progress dan Your Tasks, yang mengelola tugas-tugas harian mereka. Fitur lain yang penting di dashboard adalah Project Overview, yang memberikan daftar proyek aktif. Dari sini, pengguna bisa menggali lebih dalam ke Project Information dan Plot Information, di mana detail terkait proyek dan plot di lapangan dapat diakses.

Pengguna juga memiliki akses untuk menambahkan data plot baru melalui fitur Add Record, dan melakukan sinkronisasi data secara manual atau otomatis dengan fitur Manual Synchronize dan Allow Auto Synchronize yang berada di pengaturan. Selain itu, mereka dapat melihat Project History dan melakukan perubahan kata sandi melalui menu Settings. Pengguna juga dapat keluar dari aplikasi dengan fitur Logout.

3.2.3 Pembuatan Mockup UI Aplikasi DMRV

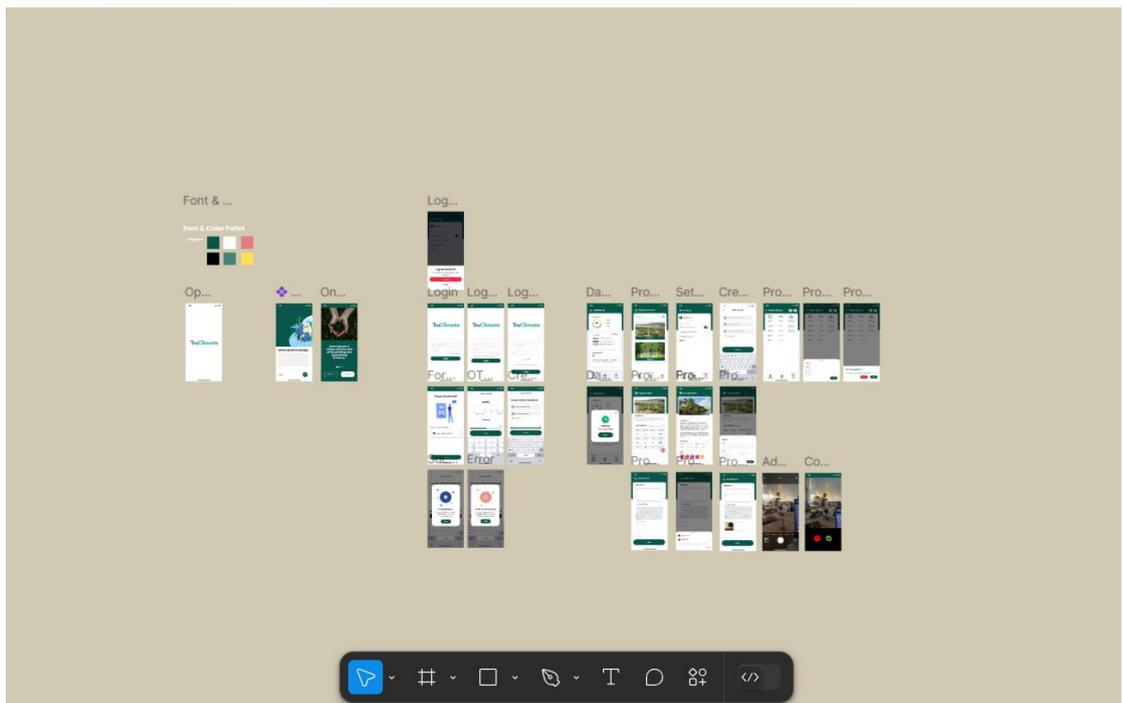
Pada tahap ini, desain antarmuka pengguna (UI) diciptakan untuk memberikan gambaran visual tentang bagaimana aplikasi akan terlihat dan berfungsi. Proses ini bertujuan untuk memastikan bahwa setiap elemen antarmuka dirancang dengan mempertimbangkan kebutuhan pengguna, kemudahan navigasi, dan pengalaman pengguna yang optimal.

Alat utama yang digunakan dalam proses ini adalah Figma, sebuah perangkat lunak desain berbasis *cloud* yang populer di kalangan desainer UI/UX. Figma memungkinkan kolaborasi tim secara *real-time*, sehingga setiap anggota tim dapat memberikan umpan balik langsung pada desain yang sedang dikembangkan. Selain itu, Figma tidak hanya memfasilitasi pembuatan desain visual, tetapi juga mendukung pembuatan prototipe interaktif yang dapat digunakan untuk menguji alur aplikasi sebelum tahap pengembangan dimulai.

Dengan menggunakan Figma, desain dan prototipe aplikasi DMRV dapat disesuaikan dengan kebutuhan proyek secara efisien, memungkinkan perbaikan dan penyesuaian yang cepat sebelum masuk ke tahap pengkodean. Hal ini penting untuk meminimalkan perubahan mendasar selama tahap pengembangan dan memastikan bahwa aplikasi dapat dikembangkan sesuai dengan visi awal.



Gambar 3. 11 Aplikasi Figma



Gambar 3. 12 Prototype Figma Aplikasi DMRV

3.2.3.1 Splash Screen

Tahap pertama dalam pembuatan mockup aplikasi DMRV adalah pembuatan splash screen. *Splash screen* merupakan layar

awal yang ditampilkan saat aplikasi pertama kali dibuka. Ini bertujuan untuk memberikan kesan pertama kepada pengguna dengan menampilkan logo dan identitas merek, dalam hal ini TruClimate, sebelum masuk ke layar utama aplikasi. Selain memperkuat branding, splash screen juga memberikan waktu bagi aplikasi untuk memuat data awal di latar belakang. Pada tahap ini, desain harus sederhana namun tetap menarik, dengan fokus utama pada elemen visual perusahaan seperti logo dan warna yang konsisten dengan tema keseluruhan aplikasi.



Gambar 3. 13 Splash Screen TruClimate

3.2.3.2 Onboarding

Gambar ini merupakan Onboarding Screen 1 dari aplikasi DMRV, yang dirancang untuk memberikan pengenalan kepada

pengguna baru tentang fitur utama aplikasi. Pada bagian atas layar, terdapat ilustrasi yang menggambarkan dunia, mencerminkan tema lingkungan global dan tindakan keberlanjutan yang relevan dengan misi aplikasi dalam memantau dan melaporkan data emisi serta perubahan iklim. Di bagian bawah, terdapat teks placeholder "*Lorem Ipsum is simply,*" yang nantinya akan diisi dengan penjelasan lebih jelas mengenai manfaat aplikasi. Selain itu, terdapat indikator halaman di bagian bawah yang menunjukkan bahwa ini adalah layar pertama dari beberapa layar onboarding, dan pengguna dapat melanjutkan ke layar berikutnya dengan menekan tombol panah di sudut kanan bawah. Desain ini menggabungkan elemen visual yang menarik dan teks deskriptif untuk menjelaskan fungsi aplikasi dengan cara yang informatif dan mudah dipahami.



Gambar 3. 14 Onboarding Screen 1



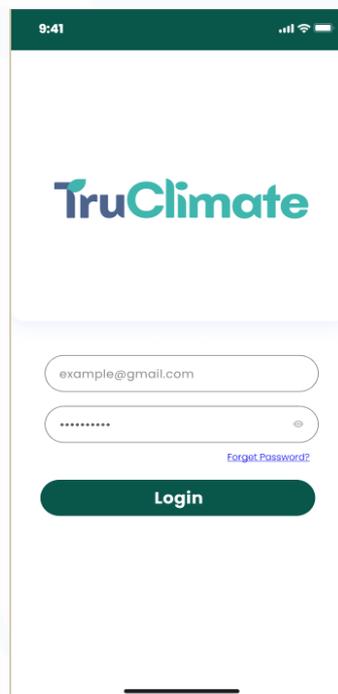
Gambar 3. 15 Onboarding Screen 2

Gambar ini merupakan *Onboarding* Screen 2 dari aplikasi DMRV, yang melanjutkan pengenalan fitur-fitur utama aplikasi. Pada layar ini, terdapat gambar visual tangan yang memegang tanah dengan tanaman kecil, yang menggambarkan tema keberlanjutan dan peran manusia dalam menjaga lingkungan, sesuai dengan misi aplikasi DMRV dalam pengelolaan proyek terkait perubahan iklim dan emisi karbon. Di bawah gambar, terdapat teks placeholder "*Lo-rem Ipsum*," yang nantinya akan diisi dengan deskripsi lebih rinci tentang aplikasi. Layar ini juga dilengkapi dengan opsi untuk pengguna mendaftar (*Register*) atau masuk (*Login*), dengan dua tombol besar di bagian bawah, memberikan pengguna pilihan untuk

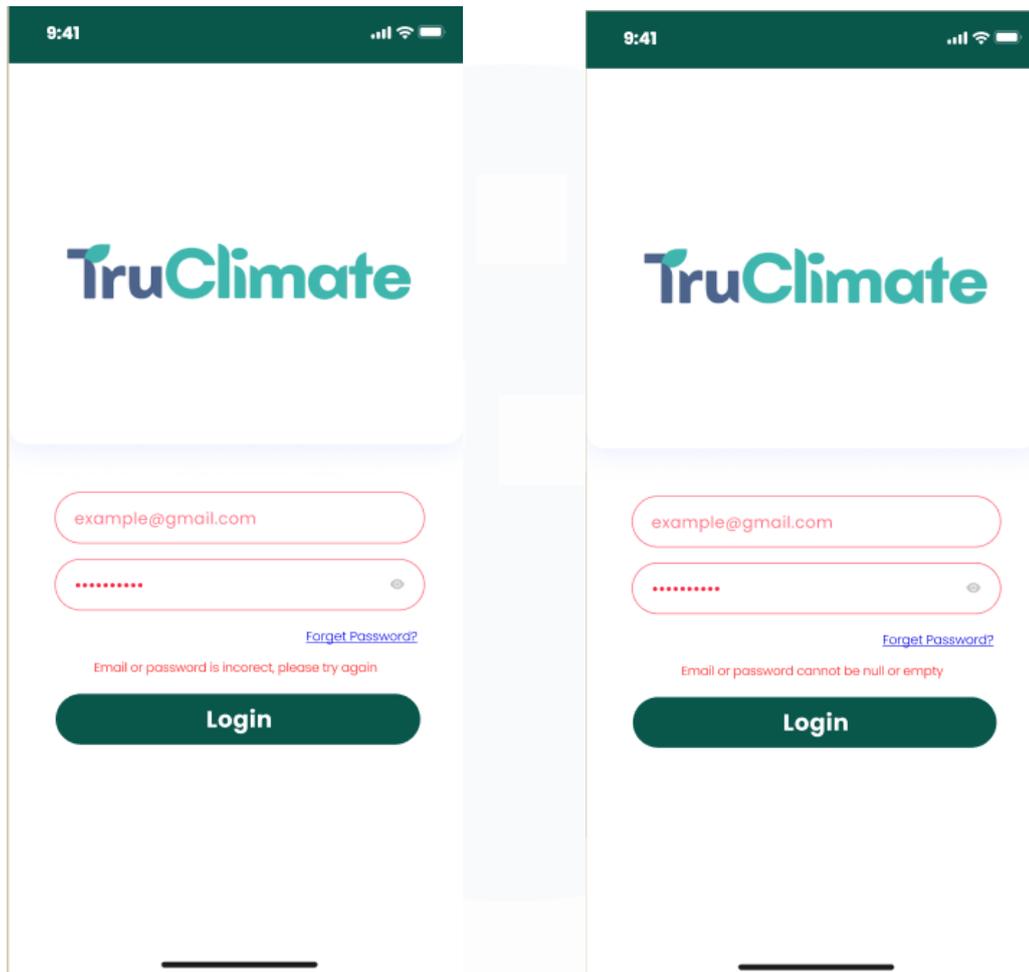
memulai menggunakan aplikasi. Desainnya sederhana dan intuitif, memadukan visual yang kuat dengan navigasi yang jelas untuk memudahkan pengguna dalam memahami tujuan aplikasi.

3.2.3.3 Login

Gambar pertama menampilkan halaman login dari aplikasi yang sepertinya bernama TruClimate, berdasarkan logo yang terlihat di bagian atas. Halaman ini merupakan mockup yang dibuat dalam Figma untuk menampilkan proses login ke aplikasi dengan mengisi *email* dan *password*. Di tengah layar, terdapat *form input* untuk *email* dan *password* dengan desain border merah. Pengguna juga diberikan opsi untuk melihat atau menyembunyikan password melalui ikon mata di sebelah kanan kolom password. Di bawah form, terdapat link "*Forget Password?*" dan tombol "*Login*" berwarna hijau gelap yang cukup mencolok.



Gambar 3. 16 Login Screen



Gambar 3. 17 Case Ketika Login Error

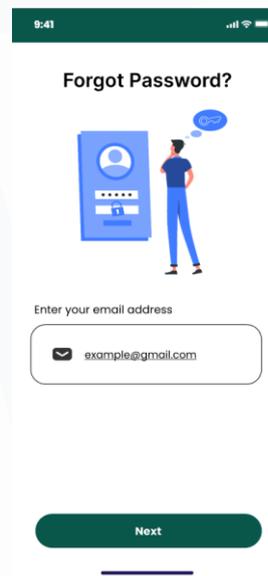
Pada gambar kedua, diperlihatkan situasi ketika pengguna memasukkan email atau *password* yang salah. Pesan kesalahan muncul di bawah form input dengan teks "*Email or password is incorrect, please try again,*" yang memberi tahu pengguna bahwa kredensial yang dimasukkan tidak valid dan harus diperiksa ulang.

Sedangkan pada gambar ketiga, situasi yang ditampilkan adalah ketika pengguna mencoba untuk login namun salah satu atau kedua kolom email dan *password* tidak diisi. Pesan kesalahan berbunyi "*Email or password cannot be null or empty,*" yang memperingatkan pengguna bahwa mereka harus mengisi semua kolom yang diperlukan sebelum dapat melanjutkan proses *login*. Mockup ini secara keseluruhan

menunjukkan alur validasi yang jelas untuk skenario login, penanganan error saat informasi yang diberikan tidak valid, serta antarmuka yang ramah pengguna.

3.2.3.4 Forget Password

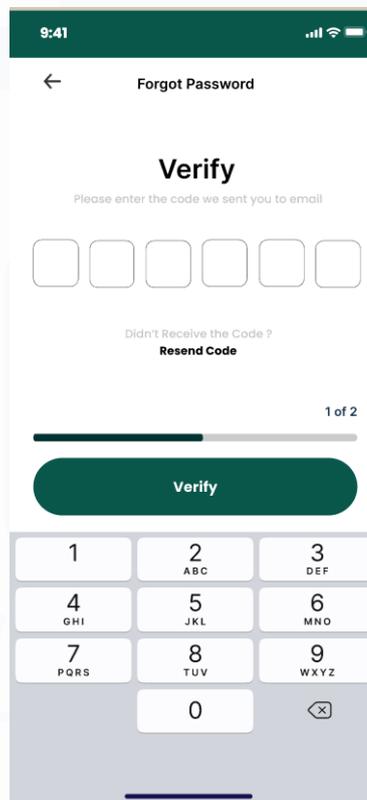
Pada *Forgot Password Screen* di aplikasi DMRV, pengguna akan diminta untuk memasukkan alamat email yang terdaftar sebagai pengguna aplikasi. Setelah pengguna memasukkan email dan menekan tombol "Next," sistem akan mengirimkan kode OTP (*One-Time Password*) ke email yang telah dimasukkan. OTP tersebut akan digunakan untuk verifikasi dalam proses pengaturan ulang kata sandi. Selain itu, di halaman ini juga terdapat gambar yang mewakili situasi lupa kata sandi, yang memperkuat konteks layar ini dan memberikan visualisasi yang relevan bagi pengguna. Desain ini dirancang untuk memandu pengguna dengan jelas dalam memulihkan akses ke akun mereka dengan langkah-langkah yang mudah dipahami.



Gambar 3. 18 Forgot Password Screen

3.2.3.5 Verify OTP Forgot Password

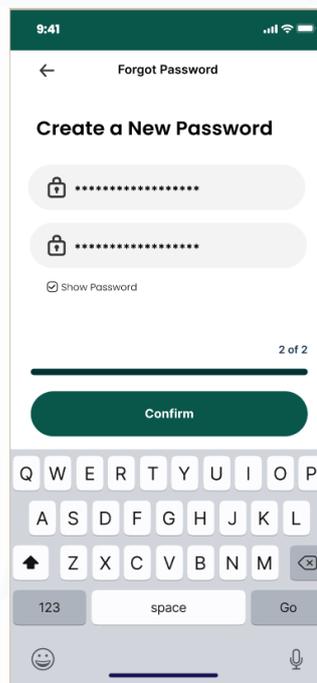
Pada *Verify OTP Forgot Password Screen* di aplikasi DMRV, pengguna akan diminta untuk memasukkan 6 digit kode OTP yang telah dikirimkan melalui email. Layar ini berfungsi sebagai langkah verifikasi sebelum pengguna dapat melanjutkan proses pengaturan ulang kata sandi. Jika pengguna tidak menerima kode OTP atau jika kode yang diterima sudah kadaluarsa, terdapat opsi "*Resend Code*" yang dapat dipilih untuk mengirimkan ulang kode OTP ke email pengguna. Desain ini dibuat untuk memastikan keamanan tambahan dalam proses pemulihan akun, sekaligus memberikan fleksibilitas kepada pengguna jika mereka menghadapi masalah dengan kode OTP yang dikirimkan. Sistem juga mengingatkan pengguna untuk memeriksa kotak masuk *email* mereka dengan seksama atau bagian spam jika OTP belum diterima dalam beberapa saat.



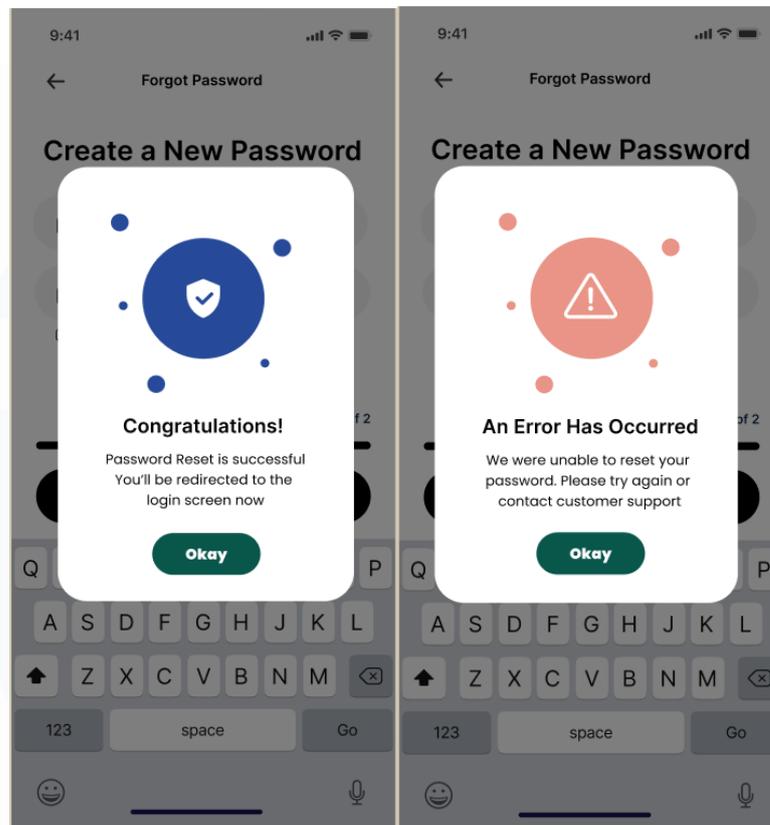
Gambar 3. 19 Verify OTP Forgot Password Screen

3.2.3.6 Create New Password

Pada *Create New Password Screen* di aplikasi DMRV, pengguna akan diarahkan ke halaman ini setelah berhasil memasukkan kode OTP dengan benar di layar sebelumnya. Di sini, pengguna diminta untuk membuat kata sandi baru dengan mengisi dua kolom: "*Password*" dan "*Confirm Password*", untuk memastikan bahwa kata sandi yang dimasukkan sesuai. Selain itu, terdapat opsi checkbox "*Show Password*" yang memungkinkan pengguna melihat karakter yang mereka ketikkan, memudahkan untuk memastikan tidak ada kesalahan pengetikan. Setelah kata sandi dan konfirmasi kata sandi diisi dengan benar, pengguna dapat menekan tombol "Confirm" untuk menyelesaikan proses pengaturan ulang kata sandi. Fitur ini penting untuk memastikan keamanan akun pengguna, dengan memberikan kesempatan untuk membuat kata sandi yang baru dan lebih aman.



Gambar 3. 20 Create New Password Screen



Gambar 3. 21 Alert Success Gambar 3. 22 Alert Faield

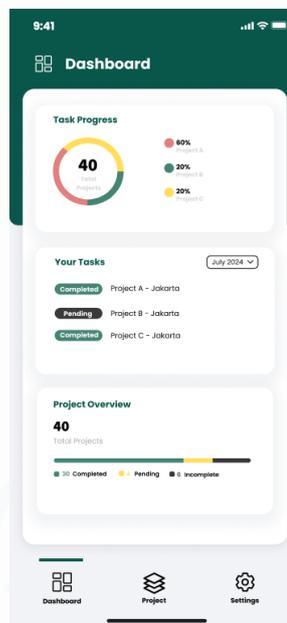
Pada Gambar 3.21, ditampilkan sebuah dialog box yang muncul ketika proses pengaturan ulang kata sandi berhasil dilakukan. Di dalam dialog box ini, terdapat *icon/picture* success yang menandakan bahwa proses reset *password* telah berhasil. Pesan sukses ini disertai tombol "*Okay*", yang ketika ditekan oleh pengguna, akan mengarahkan mereka kembali ke Login Screen, di mana mereka dapat menggunakan kata sandi baru untuk masuk ke aplikasi DMRV. Tampilan ini memberikan *feedback* visual yang jelas dan positif kepada pengguna tentang keberhasilan aksi yang mereka lakukan.

Sementara itu, pada Gambar 3.22, ditunjukkan situasi ketika pengaturan ulang kata sandi gagal. Pada dialog box ini, terdapat *icon/picture error* yang memberikan tanda visual bahwa proses *reset* mengalami kendala. Pesan yang ditampilkan menyatakan "*We were unable to reset password*", yang memberi tahu pengguna bahwa ada

masalah dalam pengaturan ulang kata sandi mereka. Tampilan ini penting untuk memberikan pemahaman kepada pengguna bahwa ada kesalahan dan mereka mungkin perlu mencoba kembali atau memeriksa informasi yang mereka masukkan sebelumnya. Kedua tampilan ini, baik sukses maupun gagal, dirancang untuk memberikan umpan balik yang jelas kepada pengguna terkait tindakan mereka.

3.2.3.7 Dashboard

Pada tampilan Dashboard Screen aplikasi DMRV, terdapat tiga bagian utama yang memberikan informasi terkait aktivitas dan status pekerjaan pengguna. Bagian pertama adalah *Task Progress*, di mana pengguna dapat melihat jumlah total progress yang sedang diambil atau dikerjakan. Dalam bagian ini, terdapat visualisasi berupa circular progress bar yang memperlihatkan secara grafis seberapa jauh progress yang telah dicapai oleh pengguna dalam menyelesaikan tugas-tugas mereka. Tampilan ini membantu pengguna untuk secara intuitif memahami sejauh mana mereka sudah berprogress dalam tugas-tugas yang diemban.

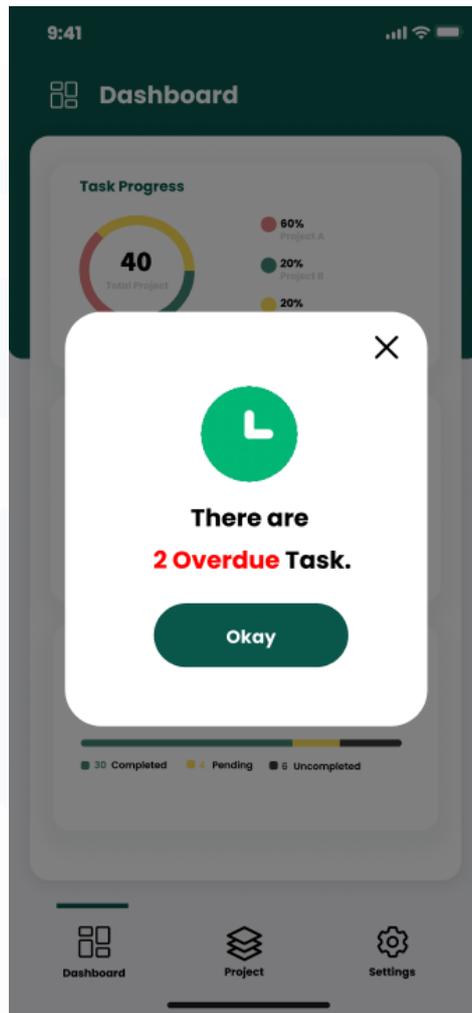


Gambar 3. 23 Dashboard Screen

Selanjutnya, bagian kedua menampilkan *Your Task*, yang menyajikan daftar tugas-tugas pengguna beserta deadline yang sedang berlangsung pada proyek yang mereka kerjakan. Pada bagian ini, pengguna juga diberi pilihan untuk melihat tugas-tugas berdasarkan bulan tertentu, sehingga mereka dapat dengan mudah melacak proyek-proyek mana yang sudah *complete* (selesai) atau pending (tertunda) dalam rentang waktu tersebut. Fitur ini memberikan fleksibilitas kepada pengguna untuk mengelola dan memprioritaskan pekerjaan mereka sesuai dengan tenggat waktu yang ada.

Bagian ketiga adalah *Project Overview*, yang menampilkan status proyek secara keseluruhan. Dalam bagian ini, progress dari setiap proyek ditampilkan secara linear, dengan status yang terbagi menjadi tiga kategori: *completed* (selesai), pending (tertunda), dan incomplete (belum selesai). Tampilan linear progress ini memberikan pandangan yang lebih rinci kepada pengguna mengenai status setiap proyek, sehingga mereka bisa memantau perkembangan dan mengetahui langkah-langkah selanjutnya yang perlu diambil untuk menyelesaikan tugas.

Secara keseluruhan, *Dashboard Screen* ini didesain untuk memberikan informasi yang komprehensif dan mudah dipahami mengenai progres kerja, tugas-tugas yang sedang berjalan, dan status proyek, sehingga pengguna dapat bekerja lebih efektif dan terorganisir.



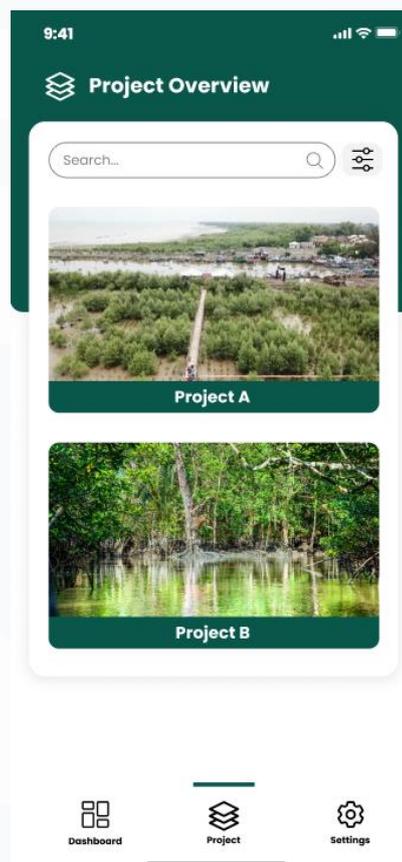
Gambar 3. 24 Alert Dashboard Screen

Selanjutnya adalah *Alert Dashboard Screen*, yang merupakan dialog screen yang muncul secara otomatis untuk memberi tahu pengguna bahwa terdapat 2 tugas (task) yang akan overdue dalam waktu dekat. Tampilan ini berfungsi sebagai pengingat penting agar pengguna segera mengambil tindakan atau menyelesaikan tugas-tugas yang sudah mendekati batas waktu. Notifikasi semacam ini sangat bermanfaat dalam aplikasi manajemen proyek seperti DMRV karena memungkinkan pengguna untuk tetap terorganisir dan terinformasi tentang tugas-tugas mereka yang mendesak. Selain memberikan notifikasi, biasanya tampilan ini juga dilengkapi dengan opsi untuk melihat lebih lanjut detail dari tugas-tugas yang overdue tersebut, sehingga pengguna dapat langsung mengakses dan menyelesaikannya. Alert

Dashboard ini dirancang untuk meningkatkan efektivitas pengguna dalam menyelesaikan pekerjaan tepat waktu, sekaligus mencegah terjadinya keterlambatan pada proyek yang sedang berjalan.

3.2.3.8 Project Overview

Pada bagian kedua dari bottom navigation yaitu *Project*, pengguna dapat melihat semua daftar proyek yang tersedia dan dapat diambil. Setiap proyek ditampilkan dalam bentuk list yang mencakup gambar serta judul proyek (*project title*), sehingga pengguna mendapatkan visualisasi yang lebih jelas mengenai proyek yang ditawarkan. Tampilan ini dibuat untuk memudahkan pengguna dalam memilih proyek yang sesuai dengan minat atau kebutuhan mereka.



Gambar 3. 25 Project Screen

Selain itu, terdapat juga search button yang memungkinkan pengguna untuk mencari proyek tertentu dengan cepat berdasarkan kata kunci yang relevan. Fitur pencarian ini membantu dalam mempersempit daftar proyek yang ditampilkan, terutama ketika ada banyak proyek yang tersedia.

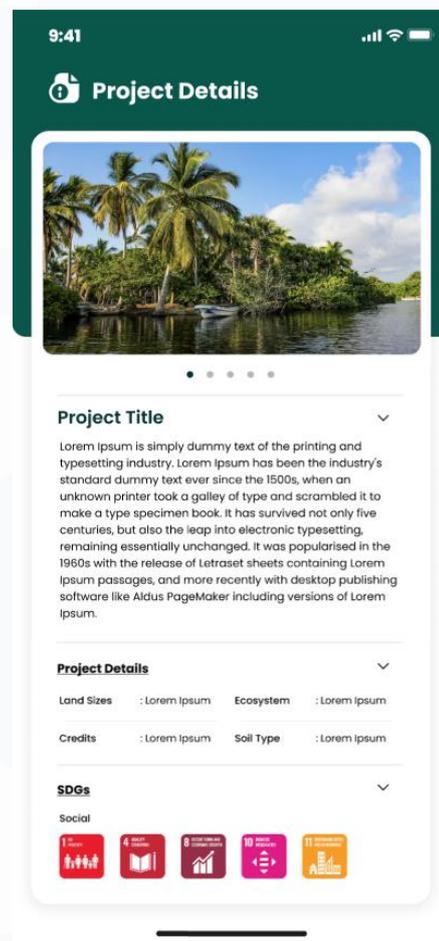
Di samping itu, ada pula filter button yang memungkinkan pengguna untuk memfilter proyek berdasarkan kriteria tertentu, seperti status, tanggal, atau kategori proyek. Fitur filter ini memberikan fleksibilitas tambahan bagi pengguna untuk menyesuaikan tampilan proyek sesuai dengan preferensi atau kebutuhan mereka, sehingga mereka dapat lebih mudah menemukan proyek yang sesuai dengan tujuan mereka.

Dengan fitur-fitur ini, layar *Project* dirancang untuk memudahkan pengguna dalam mencari, memilih, dan mengambil proyek dengan lebih efisien dan terorganisir, serta memberikan pengalaman pengguna yang lebih personalisasi.

3.2.3.9 Project Details

Berikutnya adalah *Project Details Screen* yang menampilkan rincian lengkap dari setiap proyek. Pada bagian atas layar terdapat *slider image* yang digunakan untuk menampilkan gambar-gambar yang terkait dengan proyek tersebut, memungkinkan pengguna untuk melihat visual proyek secara interaktif dengan menggeser gambar. Di bawah *slider image*, terdapat beberapa bagian penting yang dibungkus dalam *expansion tile* yang dapat dibuka atau ditutup sesuai keinginan pengguna. Bagian-bagian tersebut meliputi judul proyek, deskripsi proyek, detail proyek, dan SDG (*Sustainable Development Goals*) yang terkait. Judul proyek akan menampilkan nama proyek yang sedang dilihat, sementara deskripsi proyek menyediakan informasi latar belakang atau penjelasan umum tentang

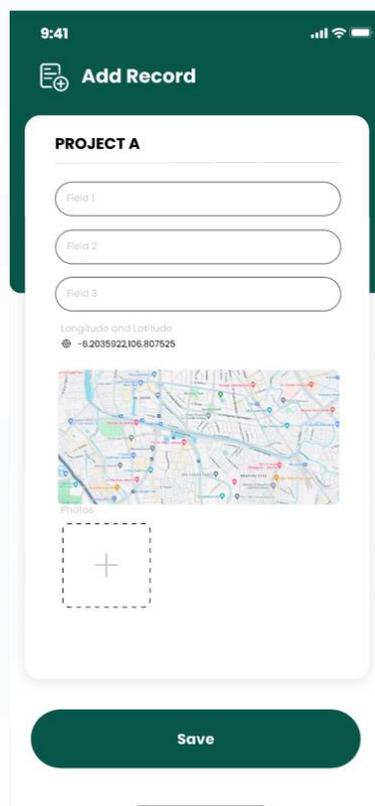
proyek tersebut. Di dalam detail proyek, pengguna dapat melihat informasi yang lebih mendalam seperti lokasi, waktu pelaksanaan, dan aspek penting lainnya. Bagian SDG menampilkan indikator-indikator yang berkaitan dengan tujuan pembangunan berkelanjutan, memberikan informasi tentang bagaimana proyek tersebut berkontribusi terhadap pencapaian SDG. Dengan adanya fitur *expansion tile* ini, pengguna dapat memilih untuk fokus pada informasi yang mereka butuhkan tanpa harus melihat semua detail secara bersamaan, sehingga antarmuka menjadi lebih intuitif dan efisien untuk mengelola informasi yang kompleks terkait proyek.



Gambar 3. 26 Project Details Screen

3.2.3.10 Add Record Project

Pada Gambar 3.27, ditampilkan layar *Add Record Project Screen* yang dirancang untuk memungkinkan pengguna menambahkan catatan proyek baru. Layar ini memiliki beberapa elemen penting. Di bagian atas, terdapat title yang memberikan keterangan tentang aktivitas yang sedang dilakukan pengguna, yaitu menambah catatan proyek. Di bawahnya, terdapat beberapa *field* yang harus diisi oleh pengguna, termasuk *location* yang terdiri dari *longitude* dan *latitude*, yang berfungsi untuk menentukan posisi geografis proyek. Untuk memudahkan pengguna, terdapat juga indikator maps yang membantu visualisasi lokasi secara lebih akurat.



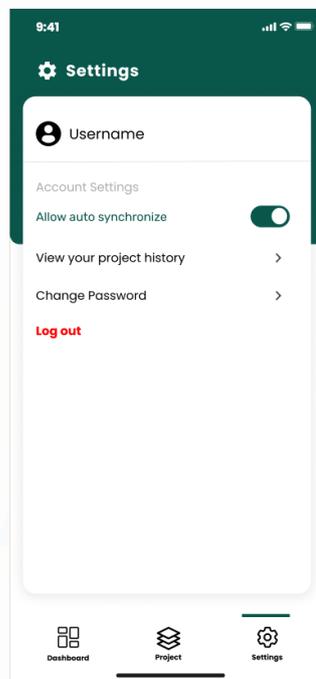
Gambar 3. 27 Add Record Project Screen

Selain itu, ada juga *field* foto yang memungkinkan pengguna mengunggah gambar terkait proyek tersebut, sehingga setiap catatan

proyek bisa dilengkapi dengan dokumentasi visual. Di bagian bawah layar, terdapat tombol *save* yang akan menyimpan seluruh informasi yang telah dimasukkan pengguna ketika ditekan. Layar ini didesain dengan antarmuka yang sederhana dan fungsional agar pengguna bisa dengan mudah menginput data dan menyimpannya untuk keperluan dokumentasi dan pelaporan proyek di aplikasi DMRV.

3.2.3.11 Settings

Pada Gambar 3.28, ditampilkan *Settings Screen* dari aplikasi DMRV yang menyediakan beberapa pilihan menu untuk pengaturan aplikasi. Pengguna dapat mengelola preferensi mereka melalui berbagai opsi yang disediakan. Salah satu menu yang ada adalah *Allow Auto Synchronize*, yang memungkinkan pengguna untuk mengatur apakah sinkronisasi data akan dilakukan secara otomatis antara perangkat dan server ketika koneksi internet tersedia. Fitur ini sangat penting untuk memastikan bahwa data proyek selalu diperbarui secara *real-time* tanpa memerlukan interaksi manual.



Gambar 3. 28 Settings Screen

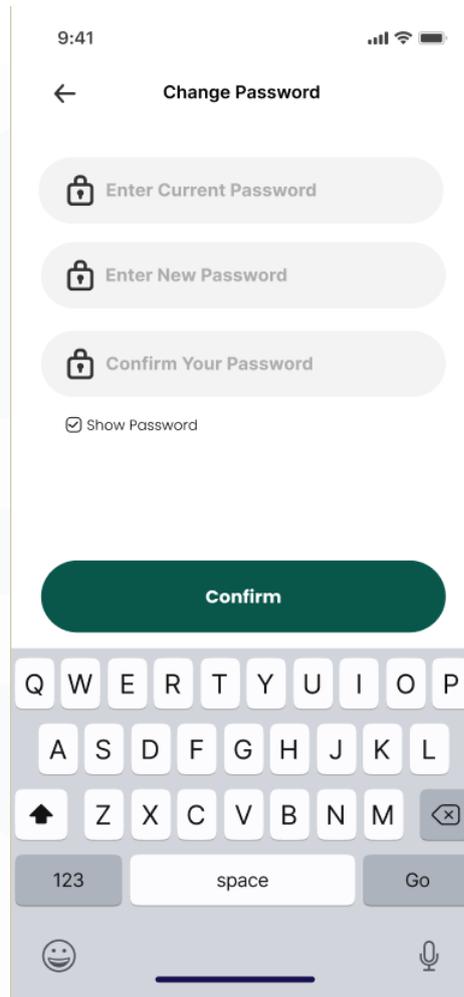
Selain itu, terdapat menu *View Your Project History* yang memungkinkan pengguna untuk melihat riwayat proyek yang telah mereka kerjakan atau ikuti. Ini memberikan kemudahan bagi pengguna untuk memantau progres dan melihat proyek-proyek sebelumnya dengan cepat.

Di dalam pengaturan ini juga terdapat opsi *Change Password*, di mana pengguna dapat memperbarui kata sandi mereka untuk menjaga keamanan akun. Ini sangat penting untuk menjaga privasi dan keamanan data sensitif yang mungkin tersimpan dalam aplikasi.

Terakhir, terdapat tombol *Logout*, yang memungkinkan pengguna untuk keluar dari akun mereka dengan aman. Fungsi ini membantu memastikan bahwa tidak ada orang lain yang dapat mengakses akun atau data pengguna ketika perangkat sedang tidak digunakan. Secara keseluruhan, *Settings Screen* ini dirancang untuk memberikan kontrol penuh kepada pengguna atas pengaturan aplikasi dan manajemen akun, sehingga mereka dapat menyesuaikan preferensi dan menjaga keamanan data mereka.

3.2.3.12 Create New Password

Selanjutnya adalah *Create New Password Screen*, yang memperlihatkan tiga *field* yang harus diisi oleh pengguna untuk memperbarui kata sandi mereka. Pada *field* pertama, pengguna diminta untuk memasukkan *current password*, yaitu kata sandi lama yang sedang mereka gunakan saat ini. Hal ini dilakukan sebagai langkah verifikasi untuk memastikan bahwa pengguna yang melakukan perubahan memang pemilik akun.



Gambar 3. 29 Create New Password Screen

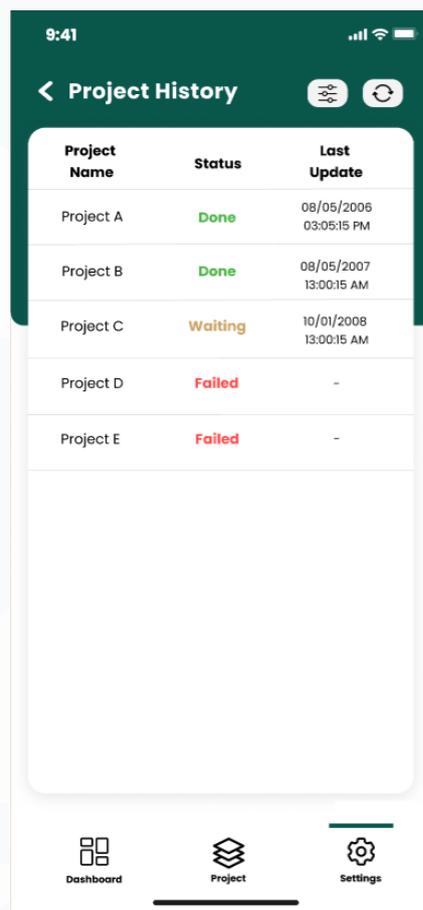
Setelah itu, pada *field* kedua, pengguna diminta untuk enter *new password*, yaitu memasukkan kata sandi baru yang ingin digunakan. *Field* ini penting untuk memastikan bahwa kata sandi baru berbeda dari yang sebelumnya dan memenuhi standar keamanan yang telah ditentukan, seperti panjang minimal atau kombinasi karakter tertentu.

Terakhir, pada *field* ketiga, pengguna perlu *confirm your password*, yakni memasukkan kembali kata sandi baru yang telah dimasukkan sebelumnya untuk memastikan tidak ada kesalahan pengetikan. Dengan adanya tiga *field* ini, layar *Create New Password* dirancang untuk memastikan bahwa proses penggantian kata

sandi dilakukan dengan aman dan benar, serta memberikan perlindungan terhadap potensi risiko keamanan seperti kesalahan input atau penggunaan kata sandi lama.

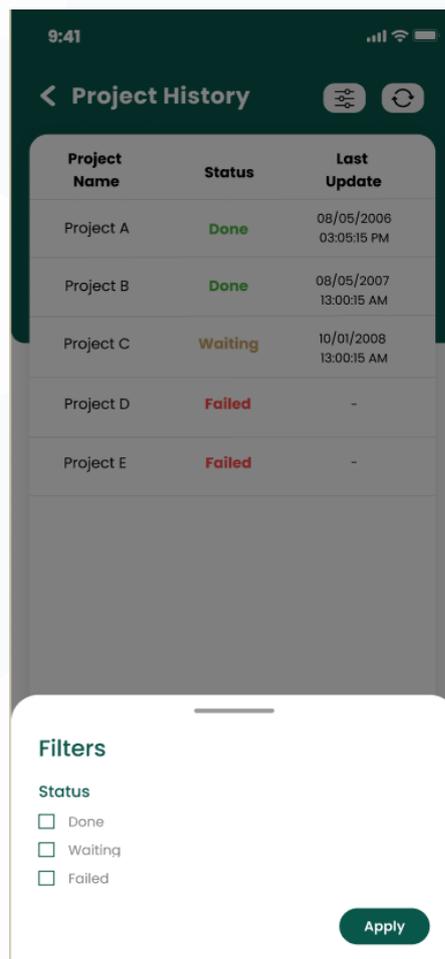
3.2.3.13 Project History

Pada *Project History Screen*, terlihat tampilan yang menampilkan sebuah data table yang berisi informasi penting mengenai riwayat proyek yang telah dikerjakan oleh pengguna. Tabel ini memuat beberapa kolom utama, yaitu *Project Name*, yang menunjukkan nama dari proyek yang tercatat dalam riwayat pengguna; *Status*, yang memberikan informasi apakah proyek tersebut sudah selesai, masih dalam proses, atau mungkin tertunda; serta *Last Update*, yang mencatat kapan terakhir kali proyek tersebut diperbarui.



Gambar 3. 30 Project History Screen

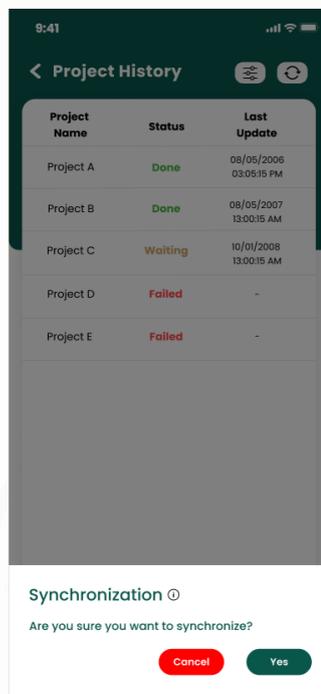
Selain itu, di layar ini juga terdapat *button filter* yang memungkinkan pengguna untuk menyaring proyek-proyek berdasarkan kriteria tertentu, seperti status atau tanggal pembaruan, sehingga memudahkan pengguna dalam menemukan informasi yang spesifik. Di samping itu, tersedia juga button untuk melakukan sinkronisasi secara manual (*manual sync*), yang memberi fleksibilitas bagi pengguna untuk memperbarui data riwayat proyek secara langsung tanpa harus menunggu sinkronisasi otomatis. Fitur ini sangat berguna terutama ketika pengguna membutuhkan data yang selalu *up-to-date* untuk keperluan monitoring atau pelaporan proyek.



Gambar 3. 31 Filter Pop Up Modal

Pada Gambar 3.31, ditampilkan modal filter yang muncul ketika pengguna menekan button filter di *Project History Screen*. Modal ini berfungsi untuk membantu pengguna menyaring proyek-proyek berdasarkan status tertentu yang ingin mereka lihat. Di dalam modal tersebut, terdapat beberapa *checkbox* yang bisa dipilih oleh pengguna sesuai dengan preferensi mereka. Pilihan yang tersedia mencakup status proyek seperti *done* (selesai), *waiting* (menunggu), dan *failed* (gagal).

Dengan fitur ini, pengguna dapat dengan mudah menyesuaikan tampilan data proyek yang ingin mereka lihat, misalnya hanya menampilkan proyek yang sudah selesai atau yang masih menunggu tindakan lebih lanjut. Fitur modal filter ini memberikan fleksibilitas dalam mengelola dan menyaring informasi proyek, sehingga memudahkan pengguna untuk fokus pada proyek-proyek yang penting atau membutuhkan perhatian khusus tanpa harus melihat seluruh daftar proyek yang ada.



Gambar 3. 32 Synchronization Modal Bottom

Pada gambar di atas, terlihat adanya *synchronization* modal bottom yang muncul ketika pengguna menekan tombol sync. Modal ini berfungsi sebagai langkah konfirmasi sebelum proses sinkronisasi data dimulai. Di dalam modal tersebut, terdapat pertanyaan yang menanyakan apakah pengguna ingin melanjutkan untuk melakukan *synchronize*, yaitu proses sinkronisasi manual yang akan memperbarui data di aplikasi dengan server.

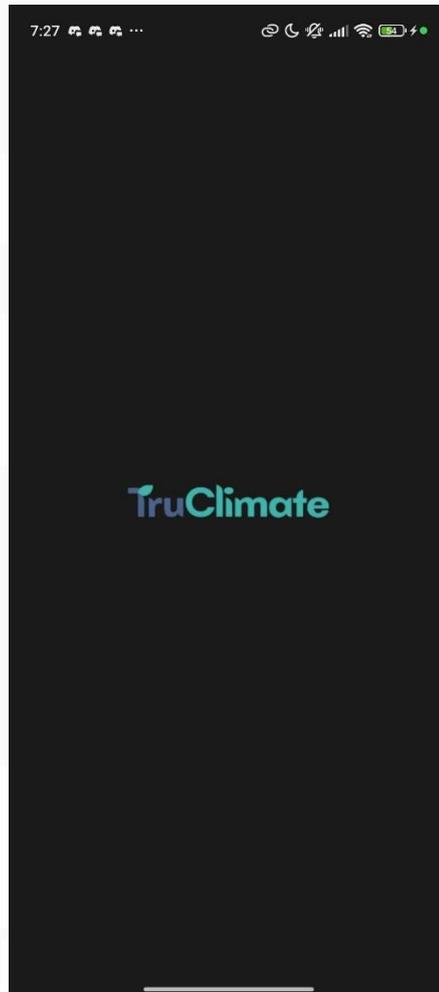
Dengan adanya modal konfirmasi ini, aplikasi memberikan pengguna kesempatan untuk memastikan bahwa mereka ingin melanjutkan proses sinkronisasi, sehingga mencegah sinkronisasi yang tidak disengaja. Selain itu, modal ini juga berfungsi untuk memberikan peringatan atau kesadaran kepada pengguna bahwa tindakan tersebut akan memperbarui data secara keseluruhan, yang mungkin membutuhkan waktu atau koneksi *internet* yang stabil. Fitur ini penting untuk menjaga integritas data dan memastikan bahwa sinkronisasi hanya dilakukan ketika pengguna merasa siap.

3.2.4 Develop Aplikasi

Setelah menentukan perangkat lunak dan *framework* yang akan digunakan, langkah selanjutnya adalah memulai pengembangan aplikasi. Proses ini akan dimulai dengan pengembangan bagian *front end*, diikuti oleh bagian *back end*. Dalam pengembangan aplikasi ini, baik *front end* maupun *back end* akan disesuaikan dengan kebutuhan dan masukan dari tim internal TruClimate untuk memastikan keselarasan dengan visi dan misi perusahaan. Selanjutnya, di bawah ini akan dijelaskan tentang tampilan aplikasi yang telah dikembangkan hingga saat ini, serta fitur-fitur yang ada dan bagaimana aplikasi tersebut dapat memenuhi kebutuhan pengguna secara efektif.

3.2.4.1 Splash Screen

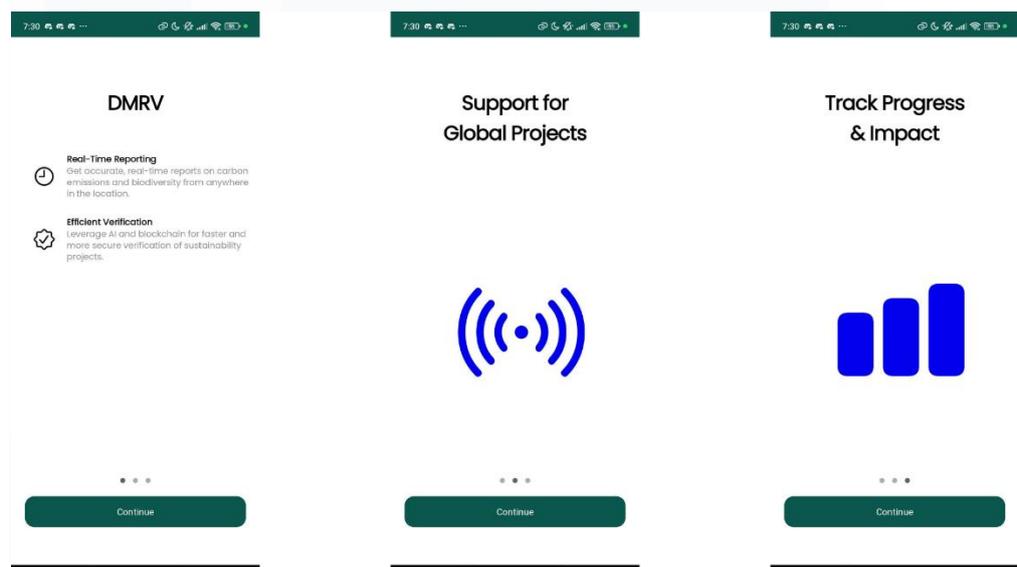
Pada gambar di atas, terlihat tampilan splash screen dari aplikasi DMRV yang muncul ketika aplikasi pertama kali dibuka. Splash screen ini menampilkan logo TruClimate, sementara latar belakangnya secara otomatis menyesuaikan dengan tema yang terpasang di perangkat pengguna. Sebagai contoh, pada perangkat yang menggunakan *dark theme*, latar belakang *splash screen* juga berubah mengikuti tema tersebut. Hal ini memberikan pengalaman visual yang konsisten dan dinamis sesuai dengan preferensi pengguna di masing-masing perangkat.



Gambar 3. 33 Splash Screen Aplikasi TruClimate

3.2.4.2 Onboarding Screen

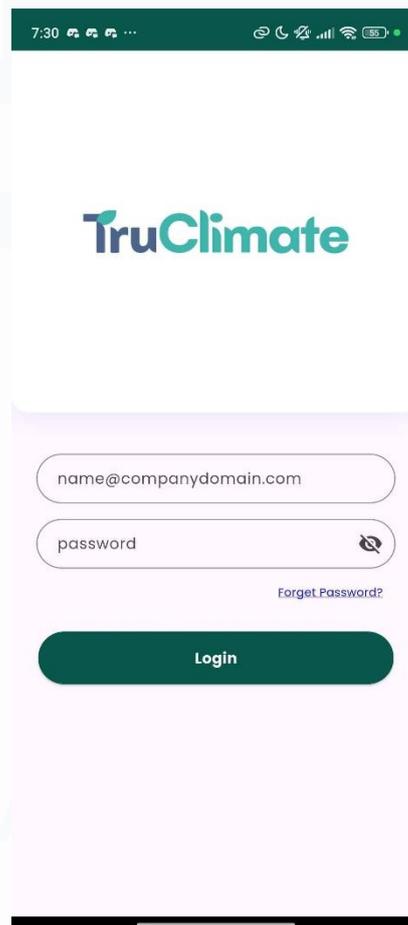
Gambar di atas menampilkan *onboarding screen* dari aplikasi DMRV, yang terdiri dari tiga *slide* yang dapat digeser oleh pengguna. *Onboarding* ini dirancang untuk memberikan gambaran umum tentang cara penggunaan aplikasi, membantu pengguna memahami fungsionalitas dasar sebelum mulai menggunakannya. Setiap *slide* memberikan informasi penting terkait fitur dan manfaat aplikasi. Setelah pengguna selesai melihat semua *slide onboarding*, layar ini tidak akan muncul lagi saat aplikasi dibuka kembali, memberikan pengalaman yang lebih efisien bagi pengguna yang sudah familiar dengan aplikasi.



Gambar 3. 34 Onboarding Screen DMRV

3.2.4.3 Login Screen

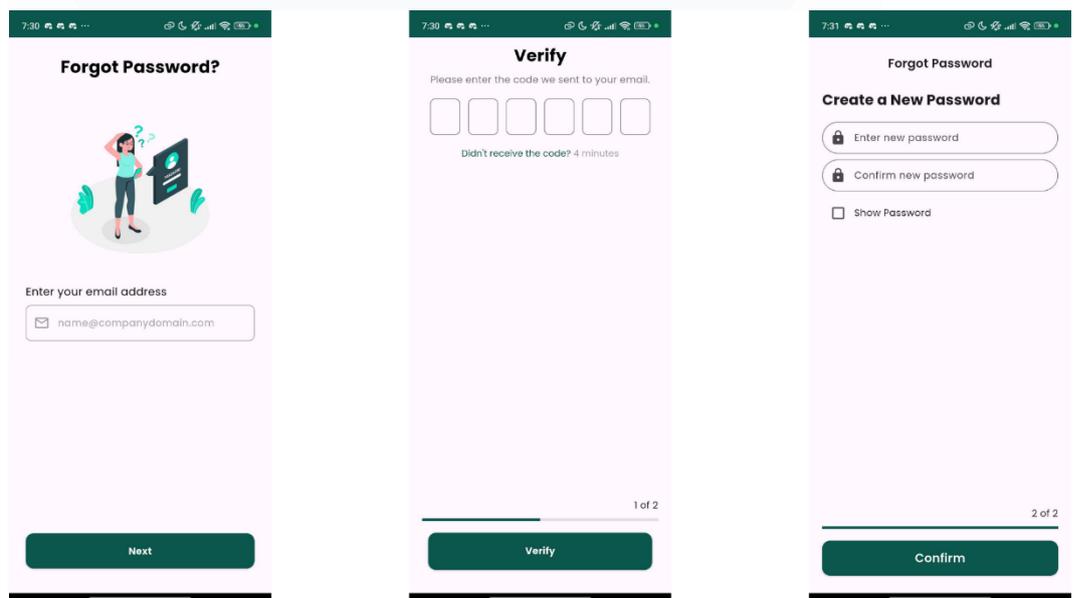
Tampilan *login screen* dalam aplikasi DMRV tidak mengalami banyak perubahan dari konsep awal yang telah dirancang dalam mockup di Figma. Namun, perlu dicatat bahwa pengguna aplikasi ini tidak dapat melakukan *registrasi* secara mandiri, karena aplikasi DMRV dirancang khusus untuk digunakan oleh staf internal dari tim TruClimate. Pengguna yang ingin mengakses aplikasi harus memiliki akun yang sudah terdaftar oleh administrator, memastikan bahwa hanya personel yang berwenang yang dapat menggunakan aplikasi ini. Hal ini bertujuan untuk menjaga keamanan dan akses eksklusif ke data yang dikelola oleh TruClimate.



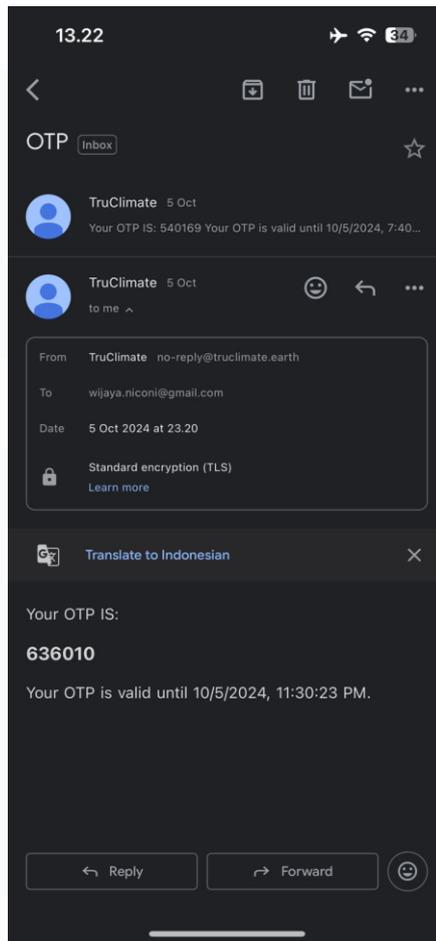
Gambar 3. 35 Login Screen

3.2.4.4 Forgot Password

Gambar di atas menampilkan *Forgot Password Screen* yang digunakan oleh pengguna yang lupa password saat ingin *login* ke aplikasi DMRV. Pada proses ini, terdapat tiga langkah yang harus diikuti oleh pengguna. Pertama, pengguna perlu memasukkan alamat email yang sudah terdaftar di *database*. Sistem akan memverifikasi apakah email tersebut sesuai dengan data yang ada. Jika email ditemukan di *database*, sistem akan melanjutkan ke langkah berikutnya, tetapi jika tidak, pengguna akan menerima pesan *error* yang menyatakan bahwa *email* tersebut tidak terdaftar di database Tru-Climate.



Gambar 3. 36 Forgot Password Screen



Gambar 3. 37 OTP Verification by Email

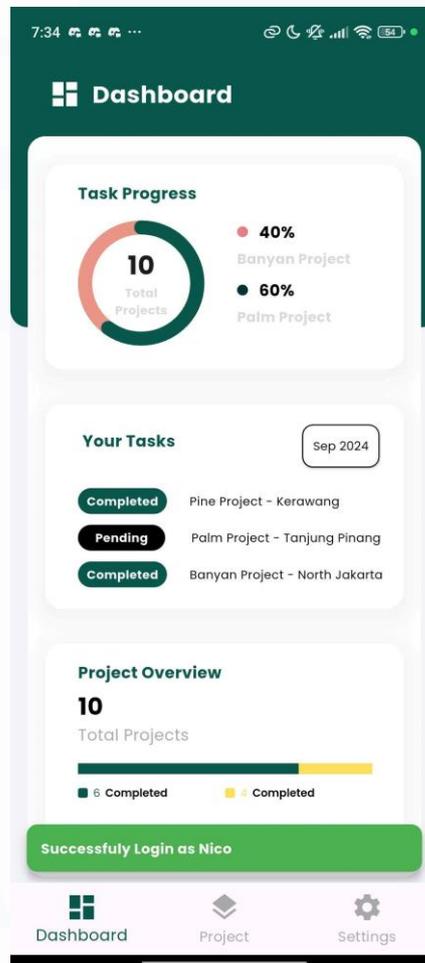
Jika verifikasi berhasil, pengguna akan menerima kode OTP yang dikirim ke *email* yang terdaftar. OTP ini harus dimasukkan dengan benar di layar berikutnya. Gambar 3.38 memberikan ilustrasi tampilan saat OTP diterima. Pengguna harus memasukkan 6 digit OTP yang sesuai dengan kode yang dikirim sebelumnya dan memastikan kode tersebut diinput sebelum masa kedaluwarsa (OTP memiliki waktu kedaluwarsa selama 10 menit untuk menjaga keamanan akun). Jika pengguna salah memasukkan OTP, akan muncul pesan *error* yang menyatakan bahwa OTP tidak sesuai.

Setelah OTP berhasil diverifikasi, pengguna akan diarahkan ke layar berikutnya untuk membuat *password* baru. Pada langkah ini, pengguna diminta untuk mengisi *password* baru dan konfirmasi

password. Setelah berhasil mengonfirmasi, sistem akan memperbarui *password* lama di *database* dengan yang baru, sehingga pengguna bisa *login* kembali dengan *password* yang baru saja diatur.

3.2.4.5 Dashboard Screen

Tampilan di atas menunjukkan *Dashboard Screen* yang muncul setelah pengguna berhasil login ke aplikasi DMRV. Di layar ini, terdapat tiga bagian utama yang dirancang untuk memberikan gambaran umum tentang proyek dan tugas yang sedang dikerjakan oleh pengguna.



Gambar 3. 38 Dashboard Screen

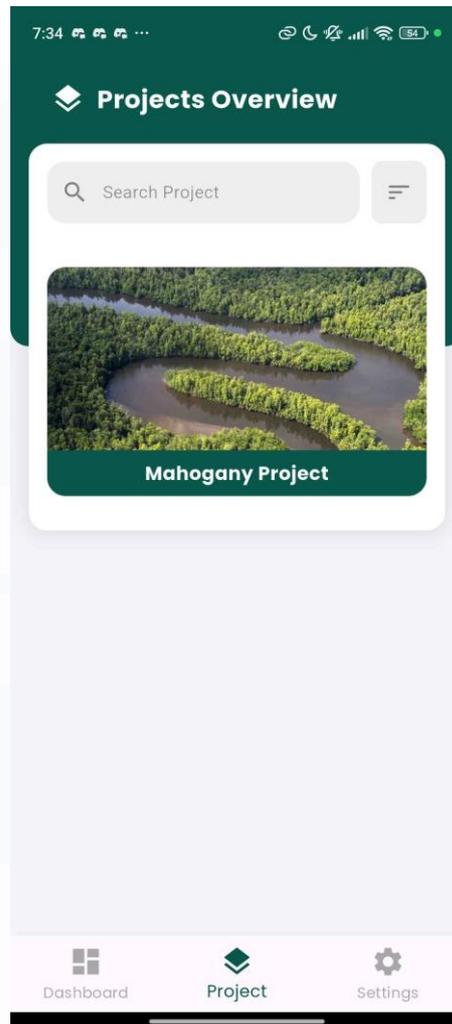
Bagian pertama adalah *Task Progress*, yang menampilkan tiga proyek utama yang sedang dikerjakan oleh pengguna. Di sini, pengguna dapat melihat jumlah total proyek yang telah diambil, dan setiap proyek ditampilkan dengan *circular progress* yang menunjukkan seberapa jauh progres dari proyek tersebut.

Selanjutnya, bagian kedua adalah *Your Task*, yang berfungsi untuk memberikan pengguna akses cepat ke daftar proyek yang masih pending atau yang telah *complete*. Pengguna dapat memilih bulan tertentu untuk melihat proyek-proyek yang sedang berlangsung atau yang telah selesai pada periode tersebut.

Terakhir, bagian ketiga adalah *Project Overview*, yang berfungsi untuk memberikan ringkasan status dari semua proyek yang sedang diambil oleh pengguna. Status proyek ditampilkan dengan jelas, menunjukkan mana proyek yang *complete* dan mana yang masih *incomplete*, sehingga memudahkan pengguna untuk melacak perkembangan pekerjaan mereka secara keseluruhan tampilan *dashboard* ini dirancang agar pengguna dapat memantau progres proyek secara efektif dan mendapatkan informasi yang dibutuhkan dengan cepat.

3.2.4.6 Project Screen

Pada layar ini, akan ditampilkan list proyek yang diambil langsung dari database TruClimate. Proyek-proyek ini ditambahkan secara manual oleh *admin* ke dalam *database*, sehingga hanya proyek yang telah diunggah oleh admin yang akan muncul di daftar ini.

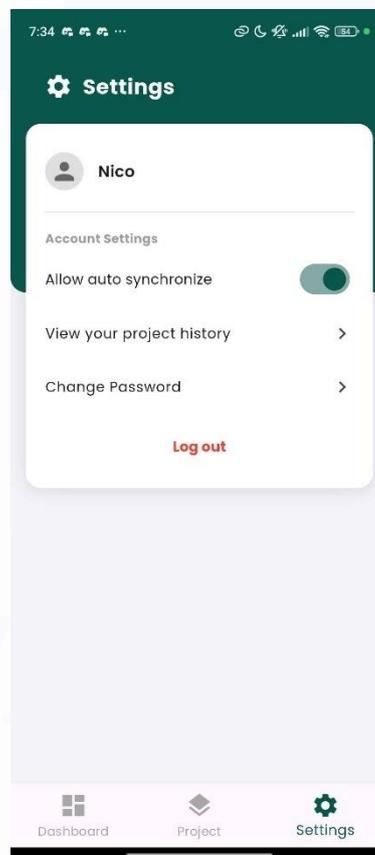


Gambar 3. 39 Project Overview

Di layar ini, pengguna dapat dengan mudah mencari proyek tertentu menggunakan *search bar* yang disediakan. Fitur ini memungkinkan pengguna untuk menemukan proyek yang mereka inginkan dengan cepat, hanya dengan memasukkan nama proyek yang ingin diambil. Hal ini memudahkan pengguna untuk menavigasi daftar proyek yang tersedia, terutama ketika jumlah proyek yang tersedia cukup banyak.

3.2.4.7 Settings Screen

Pada layar *Settings* di atas, terlihat beberapa opsi pengaturan yang memungkinkan pengguna untuk mengelola berbagai aspek aplikasi DMRV. Salah satu fitur yang tersedia adalah "Allow Auto Synchronize," yang memungkinkan aplikasi secara otomatis menyinkronkan data pengguna dengan server TruClimate, sehingga memudahkan pengguna dalam menjaga data mereka tetap terbaru. Selain itu, pengguna juga dapat melihat riwayat proyek yang telah mereka kerjakan melalui opsi "View Your Project History," yang memberikan informasi mengenai status dan perkembangan proyek mereka. Pengguna juga dapat mengubah kata sandi melalui fitur "Change Password" jika diperlukan, serta terdapat opsi untuk keluar dari aplikasi dengan memilih "Logout" untuk mengakhiri sesi penggunaan.

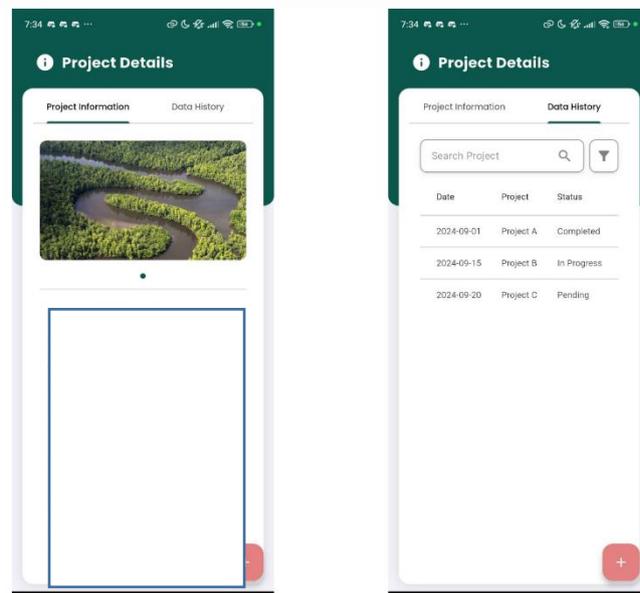


Gambar 3. 40 Settings Screen

Di bagian atas layar ini, nama pengguna ditampilkan secara jelas, yang diambil langsung dari database user TruClimate. Hal ini membantu dalam mengidentifikasi akun yang sedang digunakan dan memberikan akses yang mudah ke pengaturan-pengaturan penting yang disesuaikan dengan akun pengguna tersebut.

3.2.4.8 Project Details

Gambar di atas menampilkan *Project Details Screen*, yang terbagi menjadi dua tab utama, yaitu *Project Information* dan *Data History*. Pada tab *Project Information*, pengguna dapat melihat informasi lengkap mengenai proyek yang sedang dikerjakan. Informasi yang ditampilkan meliputi title (judul proyek), *description* (deskripsi proyek), SDG (*Sustainable Development Goals*) yang terkait, serta berbagai detail penting lainnya yang relevan dengan proyek tersebut.

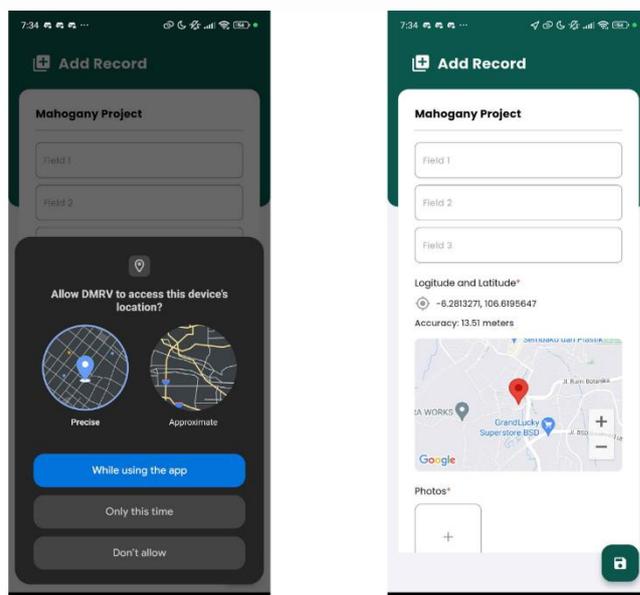


Gambar 3. 41 *Project Details*

Sedangkan pada tab *Data History*, pengguna disuguhkan dengan rekam jejak status dari proyek yang sedang berlangsung. Di sini, pengguna bisa melacak perkembangan proyek, melihat status apakah proyek sudah *completed* (selesai), *pending* (tertunda), atau bahkan *incomplete* (belum selesai). Fitur ini sangat membantu dalam memantau kemajuan proyek dan memastikan setiap tahapan berjalan sesuai rencana.

3.2.4.9 Add Record

Gambar di atas memperlihatkan *Add Record Details Screen*, di mana pengguna akan diminta memberikan izin lokasi jika sebelumnya belum diaktifkan. Jika izin lokasi tidak diberikan, maka *latitude* dan *longitude* pada layar tidak akan ter-*generate*, dan tombol *save* tidak akan bisa ditekan. Pada layar *Add Record*, pengguna diminta mengisi beberapa *field* terkait proyek, seperti tinggi pohon, lebar pohon, dan lain-lain. *Field-field* ini bersifat dinamis, artinya setiap proyek akan memiliki input yang berbeda, menyesuaikan dengan kebutuhan proyek masing-masing.

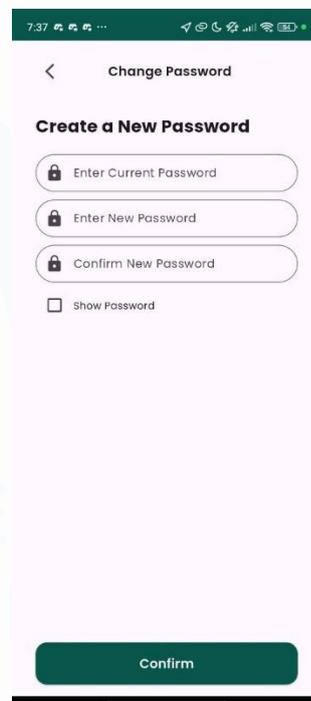


Gambar 3. 42 *Add Record Details*

Di bagian bawah *field*, ditampilkan koordinat dan tingkat akurasi lokasi pengguna, serta peta yang memperlihatkan *live location* pengguna. Selain itu, terdapat juga *field* untuk mengunggah foto, yang bisa diambil dari galeri atau diambil langsung menggunakan kamera. Setelah semua *field* yang wajib diisi (termasuk foto) sudah lengkap, pengguna dapat menekan tombol *save* untuk menyimpan data. Penting dicatat, *field* yang harus diisi dan foto bersifat *required*, sehingga harus diisi sebelum data bisa disimpan.

3.2.4.10 Change Password

Layar di bawah menampilkan *tampilan Change Password Screen*, yang muncul ketika pengguna memilih opsi *Change Password* di menu *Settings*. Pada layar ini, pengguna memiliki kesempatan untuk mengubah kata sandi lamanya dengan kata sandi baru sesuai keinginan. Pengguna diminta untuk memasukkan *current password* terlebih dahulu, lalu mengisi *new password* dan *confirm password* untuk memastikan kesesuaian input.

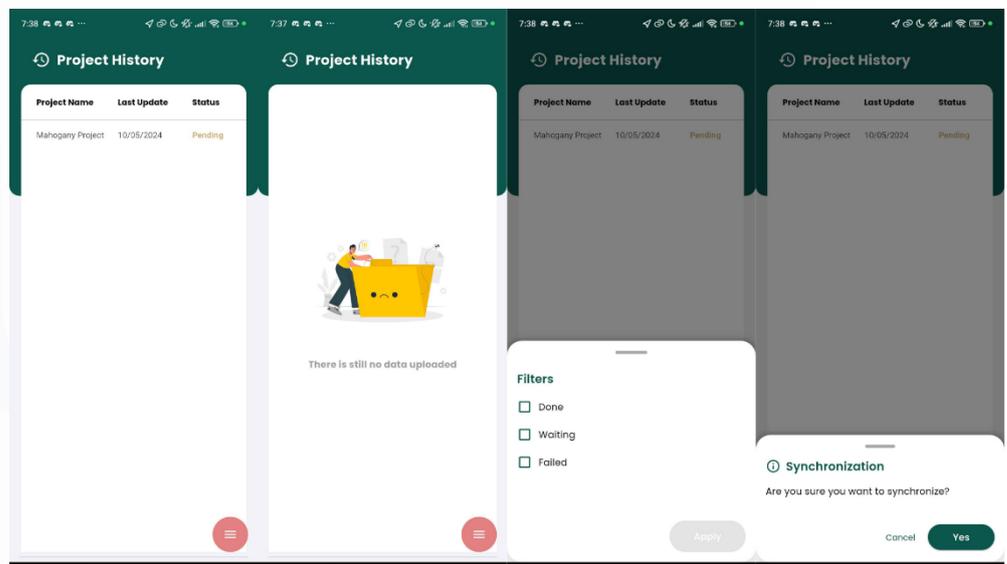


Gambar 3. 43 *Change Password Screen*

Setelah semua *field* diisi dengan benar, pengguna dapat menekan tombol *Confirm*. Ketika tombol ini ditekan, sistem akan secara otomatis menggantikan kata sandi lama dengan kata sandi baru yang telah dimasukkan, dan perubahan ini akan segera berlaku. Fitur ini dirancang untuk meningkatkan keamanan akun, sekaligus memberikan fleksibilitas bagi pengguna untuk memperbarui kata sandi mereka kapan saja diperlukan.

3.2.4.11 Project History

Gambar 3.44 menampilkan *Project History* di aplikasi DMRV. Pada gambar pertama, terlihat daftar proyek yang telah disimpan sebelumnya saat pengguna menggunakan fitur *Add Record*. Setiap proyek memiliki status yang menunjukkan apakah proyek tersebut masih pending atau sudah *done*. Status pending berarti data proyek yang ditambahkan melalui *Add Record* masih tersimpan secara lokal di perangkat, dan belum tersinkronisasi dengan server karena mungkin tidak ada koneksi *internet*.



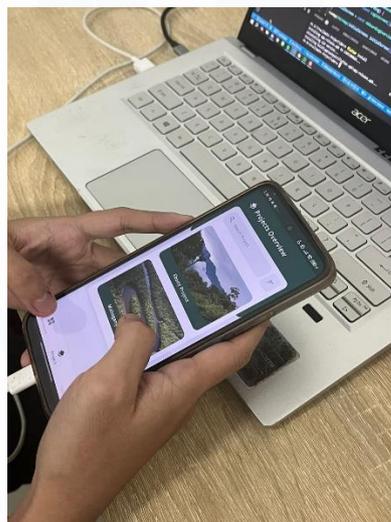
Gambar 3. 44 Project History Screen

Namun, jika fitur *Allow Auto Sync* di pengaturan diaktifkan, data akan secara otomatis tersinkronisasi begitu koneksi *internet* tersedia. Selain itu, pengguna juga bisa memilih untuk melakukan sinkronisasi secara manual dengan menekan *tombol Floating Action Button (FAB)* yang terletak di sudut kanan bawah layar.

Pada gambar kedua, terdapat fitur untuk memfilter proyek-proyek berdasarkan statusnya, seperti apakah proyek sudah terselesaikan atau belum. Jika belum ada proyek yang ditambahkan ke dalam daftar *Project History*, layar akan menampilkan pesan yang menandakan bahwa belum ada *record* proyek yang disimpan.

3.3 Evaluasi dan Testing

Proses evaluasi dan testing merupakan tahap akhir yang dilakukan setelah pengembangan aplikasi selesai. Namun, dalam kasus ini, evaluasi dan *testing* belum sepenuhnya dilaksanakan karena masih ada perubahan pada desain dan alur bisnis yang memengaruhi aplikasi. Perubahan tersebut menyebabkan penundaan dalam pelaksanaan evaluasi dan *testing*, sehingga tahapan ini harus menunggu sampai desain dan alur bisnis benar-benar *final*. Setelah semua perubahan diselesaikan, barulah proses evaluasi dan testing dapat dilakukan secara menyeluruh untuk memastikan aplikasi berfungsi sesuai dengan spesifikasi dan kebutuhan pengguna.



Gambar 3. 45 Proses Testing yang dilakukan oleh Developer

3.4 Kendala yang Ditemukan

Kendala yang ditemukan dalam proses kerja program magang sebagai berikut:

1. Masih ada beberapa kendala yang dihadapi, salah satunya adalah kurangnya pemahaman mendalam terhadap tools dan bahasa pemrograman yang digunakan dalam pengembangan aplikasi, seperti Docker, MinIO, dan Node.js. Keterbatasan ini menyebabkan proses belajar menjadi lebih lambat dan membutuhkan waktu tambahan untuk memahami cara kerja setiap tools tersebut secara optimal.
2. Selain itu, waktu yang tersedia untuk menyelesaikan proyek juga terasa terbatas. Mengingat proyek ini melibatkan banyak aspek teknis yang kompleks, dibutuhkan waktu lebih untuk memahami dan menguasai proses kerja yang ada, sehingga menyebabkan beberapa tahapan pengembangan memakan waktu lebih lama dari yang diharapkan.

3.5 Solusi atas Kendala yang Ditemukan

Berikut adalah solusi yang ditemukan untuk mengatasi kendala selama proses magang berlangsung di antara lain:

1. Untuk mengatasi kendala dalam pemahaman tools seperti Docker, MinIO, dan Node.js, solusi yang dapat diambil adalah dengan memanfaatkan dokumentasi resmi, mengikuti tutorial, dan mengambil kursus online yang berfokus pada penggunaan alat-alat tersebut. Mengalokasikan waktu khusus untuk belajar secara bertahap akan membantu mempercepat pemahaman dan keterampilan penggunaannya. Selain itu, dukungan dari rekan kerja yang lebih berpengalaman sangat berharga, karena mereka dapat memberikan bimbingan dan menjelaskan konsep-konsep yang mungkin sulit dipahami, sehingga anak magang dapat mempercepat proses pembelajaran mereka.
2. Dalam konteks pengembangan aplikasi, supervisor juga memberikan kesempatan bagi tim *backend* untuk menyediakan API, yang berfungsi sebagai endpoint. Endpoint ini sangat penting karena dapat digunakan di *frontend*

untuk mengakses data dan fungsionalitas yang diperlukan. Dengan adanya kolaborasi antara tim *backend* dan *frontend*, proses pengembangan akan menjadi lebih terkoordinasi, dan aplikasi dapat berjalan lebih efektif sesuai dengan kebutuhan pengguna.

