



Dokumentasi App OOTDiy

Anggota	<ul style="list-style-type: none">• PM dan Backend Developer : Haifan Ghani Raihan Santos (220660121202)• Frontend Developer :<ol style="list-style-type: none">1. Lusi Sugiarti Lestari (220660121084)2. Fina Nur'aeni (220660121177)• UI/UX Design :<ol style="list-style-type: none">1. Anisa Pebriyani Huslan (220660121016)2. Nida Shofwatunnisa (220660121031)
Ide	Aplikasi OOTDiy

Latar Belakang

Fashion telah menjadi bagian penting dalam kehidupan sehari-hari, terutama bagi mereka yang ingin selalu tampil *stylish* dan *up-to-date*. Tidak heran jika banyak orang terus menambah koleksi pakaian mereka untuk berbagai kesempatan, mulai dari keperluan sehari-hari hingga acara formal. Namun, seiring dengan bertambahnya jumlah pakaian yang dimiliki, banyak orang justru menghadapi tantangan dalam mengelola dan memanfaatkan koleksi pakaian mereka secara optimal. Akibatnya, mereka merasa kebingungan saat memilih pakaian dan akhirnya hanya menggunakan kombinasi pakaian yang sama secara berulang. Salah satu penyebab utama adalah kesulitan dalam memadukan pakaian (*mix and match*), sehingga mereka cenderung memilih kombinasi pakaian yang sudah dirasa aman dan praktis.

Selain itu, keterbatasan memori dalam mengingat seluruh koleksi pakaian yang dimiliki juga menjadi faktor lain. Banyak orang sering kali lupa dengan beberapa pakaian yang mereka miliki, sehingga pakaian tersebut jarang atau bahkan tidak pernah digunakan. Di sisi lain, mencoba setiap pakaian satu per satu untuk menemukan kombinasi yang cocok bukanlah solusi yang efisien. Selain memakan banyak waktu, hal ini juga melelahkan, sehingga banyak orang tetap menggunakan pakaian yang sama meskipun memiliki banyak pilihan di lemari mereka. Berdasarkan permasalahan tersebut, dibutuhkan sebuah solusi yang dapat membantu pengguna dalam mengatur dan memadupadankan pakaian secara lebih efisien.

Aplikasi OOTDiy hadir sebagai solusi inovatif untuk mengelola koleksi pakaian secara digital dan membantu memvisualisasikan kombinasi pakaian tanpa perlu mencobanya secara langsung. Dengan adanya aplikasi ini, pengguna dapat menghemat waktu dalam memilih

pakaian, memaksimalkan penggunaan pakaian yang dimiliki, dan meningkatkan kreativitas dalam berbusana.

Dalam pengembangan aplikasi ini, beberapa fitur utama yang diusulkan meliputi:

1. Login dan Logout
2. Kamera untuk Mengambil Gambar Pakaian
3. Penyimpanan Pakaian
4. Pemilihan Pakaian atau *Mix and Match* Pakaian
5. Tampilan Hasil *Mix and Match* Pakaian
6. Penyimpanan Hasil *Mix and Match* Pakaian
7. History Hasil Mix and Match Pakaian

Melalui proposal ini, diharapkan aplikasi OOTDiy dapat memberikan manfaat nyata bagi pengguna dalam meningkatkan efisiensi pengelolaan pakaian, menghemat waktu dalam memilih outfit, serta memberikan pengalaman berbusana yang lebih kreatif, praktis, dan inovatif. Dengan fitur-fitur yang telah diusulkan, aplikasi ini tidak hanya membantu pengguna dalam *mix and match* pakaian, tetapi juga memaksimalkan penggunaan koleksi yang dimiliki.

Dengan adanya justifikasi ini, pengembangan aplikasi diharapkan mendapat dukungan yang diperlukan untuk tahap realisasi dan implementasi, sehingga dapat menjadi solusi yang efektif bagi permasalahan dalam pengelolaan pakaian sehari-hari.

Justifikasi Pemilihan Teknologi

Kritea	Ionic	Flutter	React Native	Kotlin Multiplatform	MAUI
Performa	Baik untuk aplikasi berbasis web dan mobile.	Lebih cepat karena menggunakan Dart dan rendering langsung.	Tinggi	Baik, tapi lebih cocok kalau pakai kotlin asli	Tinggi, tapi lebih kompleks
Bahasa	JavaScript, HTML, CSS	Dart	JavaScript	Kotlin	C#
Learning Curve	Mudah bagi developer dengan pengalaman	Memiliki learning curve lebih tinggi	Sedang bagi developer yang terbiasa dengan	Sulit, karena butuh penyesuaian manual	Sulit, berbasis .NET Core

	web		JavaScript		
Dukungan Komunitas	Besar, banyak library tersedia	Semakin berkembang, tetapi library masih lebih sedikit dibanding Ionic	Besar, banyak library tersedia	Sedang, masih dalam tahap pengembangan	Masih dalam pengembangan
UI/UX	Menggunakan komponen web-based, fleksibel	UI lebih konsisten dan dapat disesuaikan	Native-like, tetapi membutuhkan pustaka tambahan untuk beberapa elemen	Native	Native-like, tapi lebih modern
Ukuran Aplikasi	Lebih ringan dibanding Flutter	Lebih besar karena framework bawaan	Sedang	Sedang	Besar
Multi-App Support	Mendukung pengembangan beberapa aplikasi dalam satu codebase dengan konfigurasi proyek	Memerlukan konfigurasi khusus untuk mendukung multi-app	Mendukung, pakai React Native CLI atau Expo	Bisa, tapi perlu penyesuaian manual	Bisa
Keamanan	Cukup baik, tapi bergantung pada Framework bawaan	Sangat baik dengan ekspresi bawaan	Baik, pakai Firebase Auth	Sangat baik, cocok untuk aplikasi yang butuh keamanan tinggi	Baik, pakai fitur bawaan

Berdasarkan perbandingan di atas, tim memilih Ionic karena:

1. Menggunakan HTML, CSS, dan JavaScript, yang lebih familiar bagi tim pengembang
2. Lebih ringan dibanding Flutter dalam ukuran aplikasi
3. Mendukung pengembangan web dan mobile dengan sekali kode
4. Mendukung multi-app, memungkinkan pengembangan beberapa aplikasi dalam satu codebase dengan konfigurasi proyek yang fleksibel

Namun, jika diperlukan performa lebih tinggi di masa depan, opsi migrasi ke Flutter dapat dipertimbangkan.

Metode Pengembangan

Agar proyek berjalan efektif, kami menggunakan metode Scrum, yang merupakan pendekatan Agile dalam pengembangan perangkat lunak. Scrum memungkinkan tim bekerja secara iteratif dan inkremental dengan membagi proyek menjadi beberapa sprint yang berdurasi tetap. Setiap sprint memiliki tahapan berikut:

- Sprint Planning: Perencanaan tugas yang akan diselesaikan dalam sprint di Board Trello.
- Daily Standup: Rapat harian untuk memantau perkembangan dan hambatan.
- Sprint Review: Presentasi hasil pekerjaan kepada project manager atau mentor (Dosen).
- Sprint Retrospective: Evaluasi sprint untuk meningkatkan efisiensi tim setiap minggu.

Tim proyek ini terdiri dari:

- Mentor: Bertanggung jawab mengecek setiap sprint.
- Development Team: Terdiri dari frontend, backend, dan UI/UX dengan masing-masing *jobdesk* sebagai berikut:
 1. UI/UX Design
 - Membuat IA dan *story board*
 - Membuat wireframe
 - Membuat interface aplikasi yang responsif
 2. Frontend Developer
 - Membuat flowchart
 - Mengembangkan tampilan sesuai dengan interface dari UI/UX menggunakan ionic
 - Memastikan clean code dan tidak ada bug yang menghambat
 - Memastikan semua fitur bisa berjalan
 - Fetch dengan API dari backend
 3. Backend Developer
 - Membuat ERD
 - Mengembangkan API
 - Penulisan unit test untuk fungsi dan API

Prerequisite

Untuk pengembang, dalam menjalankan aplikasi ini, beberapa persiapan dan instalasi perlu dilakukan, terutama jika aplikasi dijalankan secara lokal. Berikut langkah-langkahnya:

1. Sistem Operasi: Windows, Linux, MacOS
2. Install Node JS versi terbaru
3. NPM (Node Package Manager)
4. Instalasi dependency: npm install

Coding Conventions

a. Frontend :

1. Node JS
2. Git
3. NPM
4. Ionic Vue
5. Vite
6. Tailwind CSS
7. Emulator
8. Capacitor
9. Pinia (State Management)
10. Axios
11. Vue Router

b. Backend :

1. Flask/Jango/Node JS/python

c. UI/UX Design :

1. Figma

Link Berkaitan Dengan Proyek

Link Repository Github: <https://github.com/OOTDiy>

Link Figma: https://www.figma.com/team_invite/redeem/GvLs9yQ1Jt9OsvFEHxdogo

Link Sprint: <https://trello.com/b/GE63mu2t/ootdiy-sprint>

FigmaJam:

<https://www.figma.com/board/d4vkpS XKODoVc6dfxBYs8z/OOTDiy-FigJam?node-id=0-1&t=ra0XbOnqbg1WV7aF-1>

Backlog Process

Sprint	Milestone	Task
1 (7 - 21 Maret 2025)	Proposal Project	<ul style="list-style-type: none">- Menentukan ide dan topik proyek yang relevan dengan multi platform.- Menganalisis perbandingan teknologi multi platform.- Memilih teknologi yang paling sesuai dengan kebutuhan proyek.

2 (22 Maret - 4 April 2025)	Perancangan Aplikasi	<ul style="list-style-type: none"> - Membuat Desain tampilan yang konsisten di berbagai perangkat menggunakan Figma - Perancangan alur logika yang mempertimbangkan perbedaan platform. - Perancangan struktur database yang efisien.
3 (5 April - 2 Mei 2025)	Pengembangan Aplikasi (Front-End)	<ul style="list-style-type: none"> - Pembuatan tampilan aplikasi menggunakan Ionic - Melakukan pengujian unit terhadap komponen UI. - Penghubungan aplikasi dengan API menggunakan format data yang umum (JSON, XML).
4 (3 - 30 Mei 2025)	Pengembangan Aplikasi (Back-End)	<ul style="list-style-type: none"> - Pembuatan fungsi-fungsi program yang efisien dan dapat diakses dari berbagai platform. - Melakukan pengujian unit terhadap fungsi dan API. - Pembuatan dan koneksi database yang handal. - Pembuatan API dengan format data yang umum.
5 (31 Mei - 13 Juni 2025)	Pengujian dan Perbaikan	<ul style="list-style-type: none"> - Melakukan pengujian unit dan integrasi di berbagai platform. - Memperbaiki bug dan error yang spesifik untuk setiap platform.
6 (14 - 27 Juni 2024)	Penyelesaian dan Dokumentasi	<ul style="list-style-type: none"> - Finalisasi aplikasi dan deployment ke App Store, Play Store, atau web. - Penyusunan laporan akhir yang mendokumentasikan proses pengembangan multi platform. - Persiapan presentasi yang menarik dan informatif.