

## BAB I PENDAHULUAN

### 1.1 Latar Belakang

Seiring dengan kemajuan teknologi digital, penggunaan uang tunai dan metode pembayaran tradisional mulai digantikan oleh solusi keuangan berbasis teknologi yang dikenal sebagai teknologi finansial atau *fintech*. *Fintech* adalah sebuah inovasi di sektor jasa keuangan yang menggabungkan konsep *financial* dengan penerapan teknologi modern dalam era digital. Menurut *National Digital Research Centre* (NDRC), *fintech* merupakan istilah yang digunakan untuk merujuk pada inovasi di bidang layanan keuangan, yang menggabungkan elemen *financial* dan *teknologi* dalam upaya menciptakan solusi yang lebih *efisien* dan *modern* [1].

Salah satu penggunaan utama teknologi finansial atau *fintech* dalam transaksi layanan keuangan adalah penggunaan teknologi dompet digital atau *e-wallet*. Kemunculan *e-wallet* membuka pintu akses yang lebih luas terhadap layanan keuangan dan menyediakan *platform* yang sederhana untuk melakukan berbagai transaksi, termasuk membayar tagihan, transfer uang, dan membeli produk atau layanan. GoPay, OVO, ShopeePay, DANA, dan Flip adalah beberapa dompet elektronik yang paling populer di Indonesia.

Menurut data Statista, segmen *fintech*, yang mencakup pembayaran alternatif, memiliki nilai transaksi tertinggi per pengguna di Indonesia pada tahun 2022. Nilai rata-rata mencapai USD 92.080 pada tahun tersebut dan diperkirakan akan meningkat menjadi USD 112.300 pada tahun 2027 [2]. Namun, dalam praktik sehari-hari, transaksi layanan keuangan seringkali dilakukan secara manual.

Contoh pembayaran tunai atau pembayaran manual adalah pembayaran untuk layanan sehari-hari kepada pekerja harian yang biasanya dibagikan dengan menggunakan amplop, seperti membagikan uang kepada buruh harian, pengemudi ojek atau petugas parkir tanpa menggunakan amplop. Pembayaran untuk perawatan, seperti penggantian biaya pengobatan atau pelayanan kesehatan, harus diselesaikan langsung dengan administrasi. Pembayaran pajak, seperti pajak penghasilan, penjualan atau properti, dilakukan dengan menyerahkan jumlah yang harus dibayarkan kepada petugas dan menerima tanda terima.

Berdasarkan pembayaran manual yang telah dijelaskan sebelumnya, difokuskan pada jenis properti, yaitu pembayaran iuran kontrakan. Sebagai contoh, pembayaran iuran kontrakan dilakukan secara tunai kepada pemilik atau agen properti tanpa melibatkan transaksi elektronik. Dalam proses ini, sejumlah uang diserahkan langsung kepada pihak yang berwenang sesuai dengan perjanjian kontrakan, dan pembayaran tersebut dicatat sebagai bukti pembayaran. Menurut Badan Pusat Statistik (BPS), pada tahun 2022 terdapat sekitar 6,13% rumah tangga dengan status kepemilikan rumah sendiri dan berstatus sewa/kontrak [3].

Dengan latar belakang tersebut, maka diperlukan sebuah inovasi baru yang dapat mempermudah pembayaran iuran kontrakan bagi para penyewa melalui pemanfaatan teknologi informasi di bidang *financial technology*. Inovasi ini akan diwujudkan melalui pengembangan aplikasi *mobile* dengan menggunakan pendekatan *design thinking* yang terdiri dari tahapan-tahapan seperti *empathize*, *define*, *ideate*, *prototype*, dan *test* dengan judul **"IMPLEMENTASI DESIGN THINKING UNTUK APLIKASI PEMBAYARAN IURAN KONTRAKAN MENGGUNAKAN E-WALLET"**.

Berikut masalah penelitian yang dibahas berdasarkan latar belakang yang telah diuraikan:

1. Bagaimana aplikasi pembayaran iuran kontrakan menggunakan *e-wallet* dapat menjadi lebih efisien dan nyaman bagi pengguna dengan menggunakan metode *design thinking*? Apa saja hal-hal yang harus dipertimbangkan?

Batasan dalam proses pengembangan aplikasi pembayaran iuran kontrakan menggunakan *e-wallet* sebagai berikut :

1. Digunakan metode *design thinking* dalam perancangan tampilan antarmuka pengguna (*user interface design*) guna memahami apa yang diharapkan dari permasalahan pelanggan untuk mencari solusi yang tepat dalam menyelesaikan permasalahan yang dihadapi.
2. Memberikan pemberitahuan status pembayaran, jenis iuran dan riwayat pembayaran.

Tujuan dari penelitian ini didefinisikan sebagai berikut berdasarkan kerangka masalah yang disajikan:

1. Menyediakan *platform* di mana penyewa dapat melakukan pembayaran iuran kontrakan melalui perangkat seluler mereka menggunakan *e-wallet*.
2. Sementara pada saat yang sama mengubah cara orang berinteraksi dengan properti mereka secara finansial.

Penggunaan teknologi finansial (*fintech*) dalam pembayaran iuran kontrakan dengan metode *design thinking* memberikan manfaat sebagai berikut :

1. Membantu menyederhanakan proses administrasi dan pembayaran.
2. Kecepatan, keterjangkauan, dan tingkat keamanan yang tinggi, menjadikan *e-wallet* pilihan utama bagi individu dan bisnis dalam penggunaan teknologi finansial.
3. Penerapan pendekatan "*design thinking*" sebagai solusi yang efektif untuk memecahkan masalah yang sesuai dengan kebutuhan pengguna dengan membuat dan menguji prototipe desain.

## 1.2 Profil

Dicoding resmi diperkenalkan pada tanggal 5 Januari 2015 dengan tujuan untuk menjadi jembatan antara para *developer* di Indonesia dengan permintaan pasar yang semakin kompetitif. Dicoding bertindak sebagai *platform* pendidikan teknologi yang bertujuan untuk melahirkan sumber daya digital dengan standar global, dengan upaya untuk mempercepat kemajuan Indonesia di dunia teknologi.

Adapun struktur organisasi adalah garis penugasan resmi yang menunjukkan peran, tanggung jawab, dan hubungan di antara para anggota industri. Struktur organisasi PT Dicoding Akademi Indonesia dapat dilihat pada Gambar 1. 1 Struktur Organisasi berikut ini.





Gambar 1. 1 Struktur Organisasi

Dicoding, yang berkantor pusat di Jl. Batik Kumeli No.50, Sukaluyu, Kec. Cibeunying Kaler, Kota Bandung, Jawa Barat, juga menjalin kemitraan dengan perusahaan teknologi terkemuka di dunia. Dicoding memiliki status sebagai *Google Authorized Training Partner* dan menjalin berbagai kemitraan dengan pemangku kepentingan teknologi, perusahaan multinasional, lembaga pemerintahan, dan perusahaan dengan cakupan nasional. Dicoding juga telah berpartisipasi dalam menyelenggarakan berbagai program seperti Bangkit, Indosat Ooredoo *Digital Camp*, Lintasarta *Digischool*, Baparekras *Digital Talent*, dan *Cloud and Back-End Developer Scholarship Program with content from AWS* pada tahun 2021.

Dicoding Academy meluncurkan Program Studi Independen Bersertifikat batch 3 pada tahun 2022 dengan jalur pembelajaran Pengembang Aplikasi Android yang berlangsung dari tanggal 18 Agustus hingga 31 Desember. Proses rekrutmen dilakukan melalui platform Kampus Merdeka, dimana peserta memilih bidang atau perusahaan yang mereka minati. Penulis, pada saat itu, memilih jalur Pengembang Aplikasi Android dari Dicoding Academy. Proses rekrutmen meliputi beberapa tahapan, yaitu pengisian administrasi berkas yang mencakup surat rekomendasi dari perguruan tinggi, surat pernyataan tanggung jawab mutlak, dan *Curriculum Vitae*. Setelah lolos seleksi administrasi, peserta mengikuti tes yang terdiri dari dasar-dasar pemrograman dan tes kepribadian dengan pertanyaan dalam Bahasa Inggris. Peserta yang berhasil melewati tahap tes diberikan kesempatan untuk menerima atau menolak tawaran. Bagi peserta yang menerima tawaran, mereka menerima email pemberitahuan kelulusan seleksi dari

Dicoding atau MBKM dan diarahkan untuk memulai pembelajaran di Dicoding Academy.

Seluruh kegiatan dalam kurikulum program studi independen Dicoding Academy telah diujicobakan secara online mulai dari Agustus 2021 hingga Desember 2021 dengan agenda sebagai berikut.

Tabel 1. 1 Timeline Kegiatan

<b>Week</b>	<b>Materi</b>
Minggu ke-1	On Boarding Preparation Class
Minggu ke-2	On Boarding Preparation Class
Minggu ke-3	Memulai Pemrograman Dengan Kotlin (s/d Rangkuman Kotlin Object-Oriented Programming)
Minggu ke-4	Memulai Pemrograman Dengan Kotlin (s/d selesai) Belajar Fundamental Aplikasi Android (s/d Knowledge Check : Fundamental) ILT Soft Skill 1
Minggu ke-5	Belajar Fundamental Aplikasi Android (s/d Submission1 : Aplikasi GitHub User) ILT Technology 1
Minggu ke-6	Belajar Fundamental Aplikasi Android (s/d Modern Android Development) ILT Soft Skill 2
Minggu ke-7	Belajar Fundamental Aplikasi Android (s/d Submission 2 : Aplikasi GitHub User (Navigation dan API) ILT Technology 2
Minggu ke-8	Belajar Fundamental Aplikasi Android (s/d Latihan Repository & Injection) ILT Soft Skill 3
Minggu ke-9	Belajar Fundamental Aplikasi Android (s/d selesai) ILT Technology 3
Minggu ke-10	Belajar Prinsip Pemrograman SOLID (s.d. selesai) Belajar Dasar UX Design (s/d Latihan Melakukan Survei) ILT Softskill 4
Minggu ke-11	Belajar Pengembangan Aplikasi Android Intermediate (s/d Pengantar Localization dan Accessibility) ILT Technology 4

Minggu ke-12	Belajar Dasar UX Design (s.d. Latihan Usability Study) Belajar Pengembangan Aplikasi Android Intermediate (s/d Latihan MediaPlayer) ILT Softskill 5
Minggu ke-13	Belajar Pengembangan Aplikasi Android Intermediate (s/d Submission Aplikasi Story App) ILT Technology 5
Minggu ke-14	Belajar Pengembangan Aplikasi Android Intermediate (s/d Teori Migrasi Database Room) ILT Softskill 6
Minggu ke-15	Belajar Pengembangan Aplikasi Android Intermediate (s/d Submission Akhir Aplikasi Story App : Kirim dan Review) ILT Technology 6
Minggu ke-16	Working on Capstone Capstone Mentoring ILT Soft Skill 7
Minggu ke-17	Working on Capstone Capstone Mentoring
Minggu ke-18	Working on Capstone Capstone Mentoring
Minggu ke-19	Working on Capstone Capstone Mentoring
Minggu ke-20	Capstone Judging
Minggu ke-21	Capstone Judging
Minggu ke-22	Transcript

Proses pembelajaran di jalur Pengembang Aplikasi Android dimulai pada tanggal 18 Agustus hingga 31 Desember 2022. Pembelajaran dilakukan secara daring dengan menggunakan dua metode, yaitu pembelajaran *asinkronus* dan belajar mandiri melalui platform Dicoding. Dalam kurikulum jalur Pengembang Aplikasi Android, peserta mempelajari sembilan topik wajib, termasuk Pengenalan ke Logika Pemrograman, Memulai Dasar Pemrograman Untuk Menjadi Pengembang Software, Belajar Dasar Git & GitHub, Memulai Dasar Pemrograman Dengan Kotlin, Belajar Fundamental Aplikasi Android, Belajar Pengembangan Aplikasi Android Intermediate, Belajar Prinsip Pemrograman Solid, Belajar Dasar UX Design, dan Meniti Karir Sebagai *Software Developer*. Selain program pembelajaran untuk Pengembang Aplikasi Android, Dicoding Academy juga menyelenggarakan kelas untuk mengembangkan *soft skill* yang terdiri dari tujuh tahap.



Pada akhir program, peserta diminta untuk menyelesaikan proyek akhir yang disebut *capstone project*. Proyek ini mendorong peserta untuk memecahkan masalah nyata di dunia dengan memilih satu dari delapan tema yang disediakan.

### 1.3 Landasan Teori

#### 1.3.1 Design thinking

Berbagai model *design thinking* telah muncul selama bertahun-tahun, namun akar dari pendekatan ini berasal dari Profesor John Arnold (1913–1963) yang terkenal karena membuat mahasiswa tekniknya di MIT membayangkan bahwa mereka sedang merancang produk untuk seseorang dari luar angkasa, bukan untuk rekan-rekan mereka (yang selalu menyukai desain mereka). Pada tahun 1957, Arnold kuliah di Universitas Stanford dimana dia membangun sekolah desain teknik. Salah satu muridnya, David Kelley, kemudian mendirikan firma desain terkemuka dunia IDEO dan D.school di Universitas Stanford. Model *Design Thinking* Kelley di Stanford terdiri dari 5 fase utama: (i) *Empathize*; (ii) *Define*; (iii) *Ideate*; (iv) *Prototype* dan (v) *Test* [4].



Gambar 1. 2 Tahapan Design Thinking.  
Sumber: <https://tinyurl.com/dicodingacademia>

Pada gambar 1.2 Proses desain UX dengan pendekatan *Design Thinking* meliputi 5 langkah sebagai berikut:

1. *Empathize* (Empati): Pada sesi pertama, masalah pengguna terkait produk diidentifikasi. Untuk mengumpulkan data dan memahami kebutuhan pengguna, berbagai pendekatan seperti observasi, wawancara, dan survei digunakan.
2. *Define* (Definisi): Langkah selanjutnya adalah mendefinisikan masalah utama yang ingin dipecahkan. Definisi yang sesuai mengontrol proses selanjutnya secara lebih sistematis.
3. *Ideate* (Ideasi): Dalam sesi ini, melakukan eksplorasi pengembangan berbagai ide yang dapat dijadikan solusi terhadap permasalahan yang telah ditetapkan

sebelumnya. Tujuan dari sesi ini adalah untuk mengumpulkan sebanyak-banyaknya alternatif solusi yang dapat dipertimbangkan.

4. *Prototype* (Prototipe): Prototipe merupakan proses yang bertujuan pengujian berdasarkan ide yang dibuat. Tahapan ini dibuat untuk mencerminkan proses dan pengalaman pengguna produk.
5. *Test* (Uji Coba): Sesi terakhir adalah sesi uji coba. Prototipe diuji oleh pelanggan untuk mendapatkan umpan balik yang digunakan untuk merevisi desain sebelum melanjutkan ke sesi implementasi untuk produk akhir.

### 1.3.2 *User flow*

Sebuah aplikasi membutuhkan antarmuka pengguna yang menarik dan pengalaman pengguna yang luar biasa untuk memuaskan penggunanya. Alur pengguna adalah proses yang dilalui pengguna dari awal hingga akhir untuk mencapai suatu tujuan [5]. Di Dalam alur pengguna, pengalaman pengguna yang optimal sangat penting karena aplikasi yang tidak memberikan pengalaman pengguna yang baik mungkin tidak memenuhi kebutuhan pengguna.

### 1.3.3 *Mockup low fidelity (Wireframe)*

Pembuatan *wireframe* merupakan teknik penting yang digunakan untuk merepresentasikan struktur antarmuka pengguna secara visual. *Wireframe* adalah *mock-up* dengan *fidelity* rendah (*low fidelity*) yang menunjukkan tata letak dasar dan fungsionalitas desain tanpa membahas detail yang terlalu mendalam [6]. *Wireframe* membantu desainer merancang struktur dan tata letak desain serta memastikan bahwa semua elemen yang diperlukan disertakan dalam produk akhir.

### 1.3.4 *Mockup high fidelity*

Mockup High-fidelity adalah desain interaktif yang secara umum bekerja dengan baik, terlihat dan terasa seperti desain akhir. Hal ini juga memadukan elemen visual seperti warna, font, tipografi, tata letak, dan konten ke dalam desain [7]. Mockup high-fidelity sangat berguna untuk mengevaluasi dan menyempurnakan aspek visual dan interaktif dari sebuah desain dan dapat dibuat dengan menggunakan berbagai alat pembuatan prototipe seperti Figma, Adobe XD atau Axure.

### 1.3.5 *Prototype*

*Prototype* atau prototipe dalam Kamus Besar Bahasa Indonesia bermakna model asli yang menjadi contoh. Prototipe memberikan gambaran kepada pengguna tentang



sistem yang ingin mereka buat [8]. Prototipe dikembangkan menggunakan *wireframe low-fidelity* dan *high-fidelity*. Tujuan dari pembuatan prototipe adalah untuk memastikan bahwa produk yang dikembangkan memenuhi kebutuhan pengguna.

### 1.3.6 *Unmoderated usability study*

*Unmoderated usability study* adalah suatu bentuk pengujian dimana responden mencoba produk yang diuji dan membagikan ulasan tanpa bantuan moderator [9]. Metode ini biasanya digunakan untuk menguji bagian tertentu dari produk (skenario interaksi spesifik) daripada memberikan wawasan tentang pengalaman pengguna secara keseluruhan.

### 1.3.7 *Kotlin*

Kotlin adalah bahasa pemrograman sumber terbuka yang berfokus pada JVM, Android, JavaScript, Wasm, dan Native. Kotlin dikembangkan oleh JetBrains dimulai pada tahun 2010 dan bersifat open source sejak awal. Versi resmi pertama 1.0 dirilis pada bulan Februari 2016 [10]. Kotlin cocok untuk pengembangan beragam aplikasi, termasuk server, web, dan mobile Android. Sementara itu, pengembangan Kotlin / Native memungkinkan penggunaannya di platform lain seperti desktop, mac OS, dan iOS.

### 1.3.8 *Desain UI*

UI atau antarmuka pengguna terdiri dari semua elemen yang memungkinkan seseorang berinteraksi dengan produk atau layanan [11]. Desain UI mencakup desain visual produk digital, seperti warna, tipografi, tombol, dan ikon untuk interaksi dengan pengguna serta berdampak pada pengalaman pengguna secara keseluruhan.

### 1.3.9 *Black-box testing*

*Black box test* merujuk pada evaluasi antarmuka perangkat lunak, menguji aspek-aspek mendasar sistem tanpa memperhatikan struktur logis internal. Teknik *black box testing* dapat digunakan untuk mendapatkan kondisi input yang memenuhi semua persyaratan fungsional program. Pengujian *black-box* bertujuan untuk mendeteksi kesalahan dalam kategori seperti fungsi yang salah atau hilang, kesalahan antarmuka, kesalahan dalam struktur data atau akses basis data eksternal, kesalahan perilaku atau kinerja, serta kesalahan inisialisasi dan terminasi [12].