

**OPTIMIZING CREDIT ANALYSIS FOR MICROBUSINESSES: A  
MOBILE APPLICATION FOR STREAMLINING THE CREDIT  
ANALYSIS PROCESS IN BPD DIY**

**PROPOSAL**



Diajukan oleh  
NOVIAN ANDIKA  
AZMI BADHIUZ ZAMAN

Kepada  
**DESK KMKK BANK BPD DIY  
YOGYAKARTA  
2023**

## KATA PENGANTAR

Dalam era teknologi yang semakin canggih saat ini, aplikasi mobile menjadi salah satu alat yang sangat penting dalam mempermudah akses informasi bagi masyarakat. Salah satu aplikasi yang dapat digunakan untuk mendukung proses analisis kredit usaha kecil adalah Aplikasi Analisis Kredit Mikro.

Aplikasi ini menggunakan teknologi Dart dan Flutter yang memudahkan untuk membuat 2 aplikasi sekaligus (*Android* dan *iOS*) menggunakan hanya dengan 1 *codebase* saja. untuk membantu proses analisis kredit yang lebih cepat, akurat, dan efisien. Dengan aplikasi ini, analis kredit dapat menginput dan menganalisis data debitur dengan cepat dan mudah,

Selain itu, aplikasi ini juga dilengkapi dengan fitur-fitur yang memudahkan proses analisa kredit, seperti simulasi tetap dan porsekot, monitoring status pengajuan, cetak berkas, unggah foto dan dokumen per debitur, *role-based access control* untuk (analis, *reviewer*, dan pemutus) dan *push notifications* jika pengajuan dikirim ke user lain. Aplikasi ini juga memiliki keamanan yang tinggi dengan menggunakan teknologi autentikasi dari *Firebase*.

Aplikasi Analisis Kredit Mikro merupakan solusi yang tepat bagi analis kredit dan pemohon kredit usaha kecil untuk mendukung proses analisis kredit yang lebih cepat, akurat, dan efisien.

Yogyakarta, Februari 2023

Penulis

## DAFTAR ISI

HALAMAN JUDUL.....	i
KATA PENGANTAR .....	ii
DAFTAR ISI.....	iii
BAB I PENDAHULUAN .....	5
I.1. Latar Belakang.....	5
I.2. Perumusan Masalah .....	5
I.2.1. Batasan Masalah .....	7
I.3. Tujuan Penelitian .....	7
I.4. Manfaat Penelitian .....	7
BAB II PELAKSANAAN PENELITIAN & PENGEMBANGAN .....	9
II.1. Timeline Pengerjaan.....	9
II.2. Stack Teknologi.....	14
II.3. Alat dan Bahan Penelitian .....	17
II.4. Metode Pembuatan Aplikasi.....	19
II.5. Meeting dengan <i>PIC (Person in Charge)</i> .....	20
II.6. <i>Job Description</i> (Tugas yang dikerjakan) .....	22
II.7. Analisis Biaya dan Anggaran .....	24
BAB III HASIL DAN PEMBAHASAN.....	28
III.1. Hasil Penelitian .....	28
III.2. Persiapan Menjalankan Aplikasi Mobile .....	33
III.3. Persiapan Menjalankan Aplikasi Web Dashboard.....	36
III.4. Pembahasan.....	36
III.4.1. Pembahasan #1: Autentikasi .....	38
III.4.2. Pembahasan #2: <i>Role-based access control (RBAC)</i> .....	40
III.4.3. Pembahasan #3: Struktur <i>Database</i> .....	42
III.4.4. Pembahasan #4: <i>Push Notification</i> .....	42
III.4.5. Pembahasan #5: Upload <i>Files</i> dan Foto .....	44
III.5. Fitur Fitur Aplikasi Analisis Kredit Mikro .....	44
BAB IV KESIMPULAN .....	46

IV.1. Kesimpulan .....	46
BAB V LAMPIRAN.....	47
LAMPIRAN #1 User Interface Homescreen .....	48
LAMPIRAN #2 User Interface Mangemen Debitur .....	49
LAMPIRAN #3 User Interface Upload Photo / Files .....	50

## **BAB I**

### **PENDAHULUAN**

#### **I.1. Latar Belakang**

Latar belakang masalah pemberian kredit usaha kecil di Indonesia cukup kompleks. Pemberian kredit usaha kecil sering kali dihadapkan dengan kendala dalam proses analisis kredit yang berhubungan dengan keterbatasan data dan keterbatasan sumber daya manusia. Hal ini menyebabkan proses analisis kredit menjadi tidak efisien dan lamban.

Selain itu, proses analisis kredit juga seringkali dilakukan dengan menggunakan data yang tidak terintegrasi dan belum terverifikasi. Oleh karena itu, dibutuhkan solusi yang dapat mempermudah proses analisis kredit dengan menggunakan data yang terintegrasi dan terverifikasi.

Salah satu solusi yang dapat diterapkan adalah dengan mengkonversi data excel yang digunakan dalam proses analisis kredit ke dalam aplikasi *mobile*. Dengan aplikasi *mobile*, data yang digunakan dalam proses analisis kredit dapat diakses secara real-time dan dapat diperbaharui secara berkala. Selain itu, aplikasi *mobile* juga memudahkan pemberi kredit untuk melakukan analisis kredit secara cepat dan efisien.

Aplikasi *mobile* yang digunakan untuk mendukung proses analisis kredit usaha kecil dengan mengkonversi data excel akan menjadi solusi yang tepat dalam mengatasi masalah proses analisis kredit yang lamban dan tidak efisien.

#### **I.2. Perumusan Masalah**

Permasalahan dalam topik ini meliputi kesulitan dalam mengakses informasi oleh pemberi kredit dan pemohon kredit usaha kecil, kesulitan dalam mengupdate data secara real-time, kesulitan dalam menjaga keamanan data yang digunakan dalam proses analisis kredit, kesulitan dalam mengintegrasikan data yang digunakan dalam proses analisis kredit, dan kesulitan dalam menyediakan laporan keuangan yang tepat dan akurat. Namun, solusi yang dapat diterapkan adalah dengan mengkonversi data excel yang digunakan dalam proses analisis

kredit ke dalam aplikasi *mobile*, yang diharapkan dapat mempermudah proses analisis kredit dengan menggunakan data yang terintegrasi dan terverifikasi, serta mempermudah akses informasi bagi pemberi kredit dan pemohon kredit usaha kecil.

*Approach* yang digunakan dalam penyelesaian topik ini adalah dengan mengkonversi data excel yang digunakan dalam proses analisis kredit ke dalam aplikasi *mobile*. Dengan aplikasi *mobile*, data yang digunakan dalam proses analisis kredit dapat diakses secara real-time dan dapat diperbaharui secara berkala. Selain itu, aplikasi *mobile* juga memudahkan pemberi kredit untuk melakukan analisis kredit secara cepat dan efisien.

Cara penyelesaiannya adalah:

1. Melakukan analisis terhadap kebutuhan pemberi kredit dan pemohon kredit usaha kecil dalam proses analisis kredit.
2. Mengumpulkan data yang digunakan dalam proses analisis kredit dari berbagai sumber yang terpercaya dan melakukan proses integrasi data.
3. Membuat desain aplikasi *mobile* yang sesuai dengan kebutuhan pemberi kredit dan pemohon kredit usaha kecil.
4. Melakukan pengembangan aplikasi *mobile* dengan mengkonversi data excel yang digunakan dalam proses analisis kredit ke dalam aplikasi *mobile*.
5. Melakukan pengujian dan validasi aplikasi *mobile*.
6. Melakukan implementasi aplikasi *mobile* dan melakukan pemeliharaan aplikasi *mobile*.

Aplikasi *mobile* yang dihasilkan dari cara penyelesaian diatas diharapkan dapat mempermudah proses analisis kredit dengan menggunakan data yang terintegrasi dan terverifikasi, serta mempermudah akses informasi bagi pemberi kredit dan pemohon kredit usaha kecil.

### **I.2.1. Batasan Masalah**

Berikut batasan masalah yang digunakan dalam penelitian ini:

1. Aplikasi mobile yang dikembangkan hanya digunakan untuk proses analisis kredit usaha kecil saja.
2. Aplikasi mobile yang dikembangkan hanya digunakan untuk proses analisis kredit dengan mengkonversi data excel ke dalam aplikasi mobile saja.
3. Aplikasi mobile yang dikembangkan hanya untuk sistem operasi Android saja.

### **I.3. Tujuan Penelitian**

1. Untuk mengembangkan aplikasi mobile yang dapat mempermudah proses analisis kredit dengan menggunakan data yang terintegrasi dan terverifikasi.
2. Untuk mengevaluasi efektivitas aplikasi mobile dalam mempermudah proses analisis kredit dan meningkatkan akses kredit bagi usaha kecil.
3. Untuk menyediakan solusi yang dapat mempermudah akses informasi bagi pemberi kredit dan pemohon kredit usaha kecil dalam proses analisis kredit.
4. Untuk menyediakan laporan keuangan yang tepat dan akurat yang dapat digunakan dalam proses analisis kredit.

Semua tujuan diatas diharapkan dapat di capai dengan mengkonversi data excel yang digunakan dalam proses analisis kredit ke dalam aplikasi mobile. Dengan aplikasi mobile ini diharapkan dapat mempermudah proses analisis kredit dengan menggunakan data yang terintegrasi dan terverifikasi, serta mempermudah akses informasi bagi pemberi kredit dan pemohon kredit usaha kecil

### **I.4. Manfaat Penelitian**

1. Memberikan solusi yang efisien dalam proses analisis kredit usaha kecil dengan mengkonversi data excel ke dalam aplikasi mobile.

2. Mempermudah akses informasi bagi pemberi kredit dan pemohon kredit usaha kecil dalam proses analisis kredit.
3. Meningkatkan efektivitas proses analisis kredit dengan menggunakan data yang terintegrasi dan terverifikasi.
4. Mempermudah pemberi kredit dalam melakukan analisis kredit secara cepat dan efisien.
5. Meningkatkan akses kredit bagi usaha kecil di Indonesia khususnya di Provinsi Daerah Istimewa Yogyakarta.
6. Memberikan laporan keuangan yang tepat dan akurat yang dapat digunakan dalam proses analisis kredit.

Penelitian ini diharapkan dapat memberikan manfaat yang signifikan bagi para pemakai, terutama bagi usaha kecil yang kesulitan untuk mendapatkan akses kredit. Aplikasi yang dikembangkan dalam penelitian ini diharapkan dapat membantu para pemakai dalam melakukan analisis kredit dan memberikan solusi yang tepat untuk mendapatkan kredit yang diperlukan.

Selain itu, aplikasi ini juga diharapkan dapat meningkatkan akses kredit bagi usaha kecil. Dengan adanya aplikasi ini, para pemakai dapat dengan mudah mengajukan kredit dan mendapatkan solusi yang tepat tanpa harus mengalami kesulitan dalam proses pengajuan kredit.

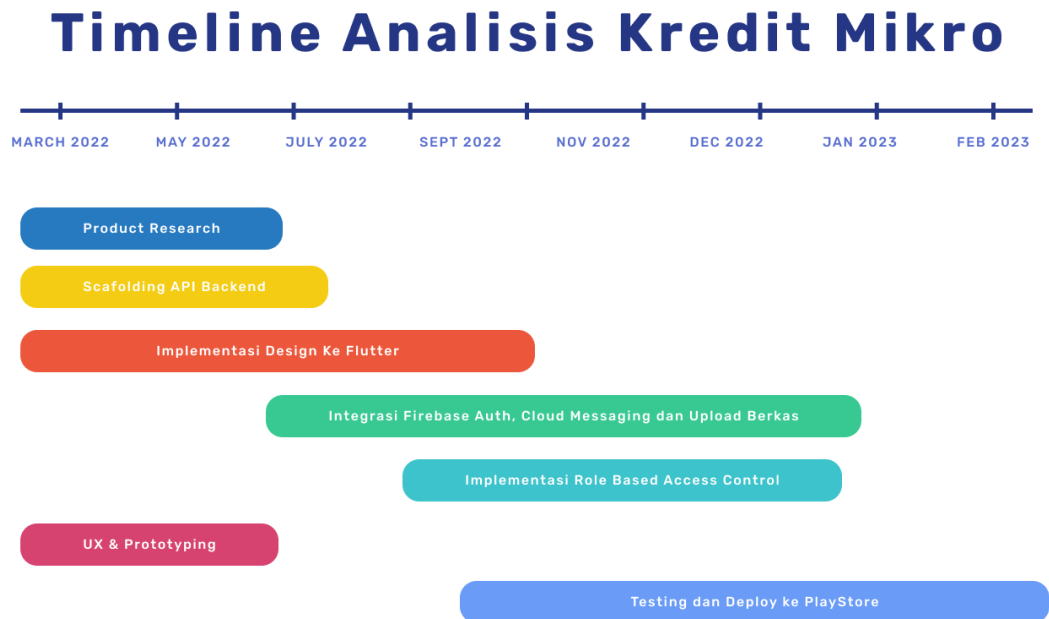
Selanjutnya, aplikasi ini diharapkan dapat memberikan kontribusi yang signifikan dalam meningkatkan efisiensi dan transparansi dalam proses analisis kredit. Aplikasi ini dapat membantu para pemakai untuk menyimpan dan mengelola data-data mereka secara efisien, sehingga dapat mempermudah proses analisis kredit. Aplikasi ini juga dapat membantu para pemakai untuk mengetahui status permohonan kredit mereka dan membuat keputusan yang lebih baik.



## BAB II

### PELAKSANAAN PENELITIAN & PENGEMBANGAN

#### II.1. Timeline Pengerjaan



**Gambar II.1.** Timeline pengerjaan aplikasi Analisis Kredit Mikro

- ***Product Research (Maret - Juli 2022)***

Disini dijelaskan tentang scope aplikasi oleh *PIC (Person in Charge)* yaitu Mas Adam dan Mas Waskhito. Dalam proyek ini, mereka akan menjelaskan dan menguraikan *scope* aplikasi yang akan dikembangkan. Selain itu, mereka juga akan memberikan berbagai contoh file excel dan file lainnya yang akan digunakan sebagai bahan referensi dan sumber informasi dalam proses penelitian dan pengembangan produk.

Kedua *PIC* akan bekerja sama dalam mengevaluasi dan menganalisis data yang dikumpulkan untuk mencapai hasil yang optimal dalam proyek ini. Mereka akan menggunakan metode yang terbaik untuk mengumpulkan data dan menganalisisnya untuk menemukan solusi yang tepat untuk permasalahan yang dihadapi. Selain itu, mereka juga akan

melakukan komunikasi yang efektif dan terbuka dengan tim yang terlibat dalam proyek ini untuk menjamin kesuksesan proyek yang akan dilakukan.

- ***Scaffolding API Backend (Maret - Agustus 2022)***

Pada rentan waktu ini, saya sedang belajar untuk memahami contoh file yang diberikan oleh *PIC*. Saya juga sekaligus mengimplementasikan pengetahuan tersebut ke dalam sebuah *code*. Hal ini dilakukan untuk mempersiapkan diri dalam proyek Product Research yang akan dilakukan.

Dalam proses pembuatan *REST API*, saya juga belajar mengenai desain struktur database, relasi antar table, dan data-data lainnya yang dikumpulkan untuk membentuk suatu *REST API* yang akan kemudian digunakan pada aplikasi *mobile*. Saya belajar bagaimana cara mengatur relasi antar table dan mengatur data yang diterima dari client agar dapat digunakan dengan baik pada aplikasi mobile yang akan dikembangkan. Ini merupakan bagian penting dalam proyek *Product Research* yang akan membantu dalam pengembangan aplikasi mobile yang akan dikembangkan.

- ***Implementasi Frontend Flutter (Maret - Oktober 2022)***

Setelah *REST API* jadi, dalam rentan waktu ini saatnya untuk mengaplikasikan kedalam tampilan *mobile* yang menarik. Dalam proses ini, saya akan mengimplementasikan rumus excel yang digunakan untuk perhitungan angsuran, laporan keuangan, dan lainnya kedalam aplikasi. Hal ini dilakukan untuk memudahkan pengguna dalam melakukan perhitungan dan menyimpan data keuangan.

Implementasi rumus excel tersebut kemudian akan di verifikasi di *server backend*. Hal ini dilakukan untuk memastikan bahwa hasil perhitungan yang ditampilkan pada aplikasi mobile sesuai dengan data yang ada di server. Proses ini membutuhkan waktu yang agak lama karena

ini adalah proses yang cukup kompleks dan memerlukan pengujian yang detail.

Dalam proses pengembangan aplikasi mobile ini, saya menggunakan framework Flutter yang baru saya pelajari. Hal ini memerlukan waktu yang juga agak lama karena saya harus belajar dari nol seperti *State Management*, *Basic Layouting*, *Perhitungan*, *Fetch Data*, *Send Data* dan lainnya.

- **Integrasi *Third Party Service* (*Firestore Auth*, *Firestore Cloud Messaging*, dan *Cloudinary*) (Juli – November 2022)**

Setelah aplikasi sudah memasuki tahap *Minimum viable product* (hitung hitungan sudah oke, sudah bisa dijalankan dan sudah di testing oleh PIC). Proyek ini akan dilakukan untuk meningkatkan keamanan dan kenyamanan pengguna dalam menggunakan aplikasi.

Dalam tahap ini, saya akan membuat autentikasi user untuk login kedalam aplikasi. Proses ini dilakukan karena aplikasi ini dipakai untuk banyak *users* yang akan menggunakannya. Oleh karena itu, proses autentikasi sangat penting untuk dilakukan untuk menjamin keamanan data pengguna dan memisahkan data antar users.

Untuk proses *autentikasi* user, saya akan mengintegrasikan *Firestore Authentication* di backend server menggunakan *Firestore Admin SDK*. Hal ini dilakukan untuk mempermudah proses *autentikasi* dan menjamin keamanan data pengguna. Selain itu, saya juga akan mengintegrasikan *push notifications* di server menggunakan *Firestore Cloud Messaging*. Hal ini dilakukan untuk memudahkan pengguna dalam menerima notifikasi penting dari aplikasi.

Dalam tahap, saya juga akan menggunakan *third party Cloudinary* untuk penyimpanan berkas. Hal ini dilakukan agar di server hanya logiknya saja yang bermain dan tidak membebani server dengan proses penyimpanan berkas yang cukup besar. Dengan menggunakan *third party Cloudinary*, proses penyimpanan berkas akan lebih efisien dan aman.

- **Implementasi *Role Based Access Control* dan Membuat Dashboard Admin untuk manajemen user (September - Desember 2022)**

Setelah semua *third party services* seperti *Firebase Auth*, *Firebase Cloud Messaging*, dan *Clouddinary* diintegrasikan ke dalam aplikasi, sekarang waktunya untuk mengatur hak akses pengguna atau yang biasa disebut *Role Based Access Control (RBAC)*. *RBAC* bertujuan untuk membatasi akses pengguna terhadap fitur atau data yang dapat diakses.

Dalam proses implementasi *RBAC*, *Firebase Auth* memiliki fitur yang memudahkan dalam hal ini yaitu dengan menggunakan *Custom Claims*. *Custom Claims* merupakan parameter yang digunakan untuk memberikan peran atau role kepada pengguna dengan mudah. Dengan *Custom Claims*, admin dapat dengan cepat dan mudah memberikan akses yang sesuai dengan peran masing-masing pengguna.

Selain itu, untuk mengatur hak akses pengguna juga dibutuhkan dashboard admin yang dapat diakses oleh role admin saja. Dalam tahap ini, saya akan membuat dashboard admin berbasis *Flutter Web* yang dapat diakses dengan menggunakan role admin saja. Dashboard ini akan menyediakan fitur untuk menambah, mengedit, dan menghapus pengguna serta mengatur hak akses pengguna sesuai dengan peran yang ditentukan.

- ***Finishing Touch, Testing, Deploying* (Desember 2022 – Februari 2023)**

Masuk ke tahap *finishing*, aplikasi sudah siap untuk digunakan oleh pengguna. Aplikasi sudah dilengkapi dengan rumus perhitungan yang dibutuhkan, integrasi *REST API* dari service yang dibutuhkan, integrasi *third party service* seperti *Firebase Auth*, *Firebase Cloud Messaging*, dan *Clouddinary*, serta *Role Based Access Control (RBAC)* yang digunakan untuk membatasi akses pengguna.

Setelah aplikasi selesai dikembangkan, dilakukan testing berkala oleh pihak *PIC* dengan menginputkan beberapa data debitur yang asli ke dalam aplikasi. Tujuannya adalah untuk memastikan bahwa aplikasi dapat

digunakan dengan baik oleh pengguna dan dapat memenuhi kebutuhan pengguna.




Agar pengguna dapat dengan mudah mendapatkan aplikasi, saya berinisiatif untuk merilis aplikasi melalui *Google Play Console*. Dengan begitu, pengguna dapat dengan mudah menemukan dan mendownload aplikasi melalui platform tersebut.



Untuk menyambungkan aplikasi *mobile* dengan *REST API*, dibutuhkan server yang dapat digunakan untuk menyimpan dan mengolah data. Disini saya menggunakan *Replit* sebagai hosting untuk *backend* aplikasi. Dengan begitu, aplikasi dapat berjalan secara *full online* dan dapat dijalankan kapan saja oleh pengguna.



Dalam proyek ini, saya juga menggunakan *Elephant SQL* sebagai *database* yang digunakan. *Elephant SQL* merupakan cloud-based *PostgreSQL*, yang berarti bahwa *database* tersebut tidak di-host di mesin fisik yang terletak di lokasi tertentu, melainkan di *cloud*. Keuntungan dari menggunakan *Elephant SQL* adalah mudah dalam *maintenance* dan *scalability*, serta dapat diakses dari mana saja yang memiliki koneksi internet. *Elephant SQL* juga menyediakan fitur *backup* dan *restore* secara otomatis, sehingga data akan tetap aman jika terjadi masalah pada *server*.

## II.2. Stack Teknologi



Berikut adalah Stack Teknologi yang digunakan pada aplikasi ini

Logo	Nama	Fungsi	Keterangan
	Dart	<i>Frontend</i>	Dart adalah bahasa pemrograman yang digunakan untuk mengembangkan aplikasi <i>mobile</i> dan web dengan menggunakan <i>framework</i> Flutter. Dart merupakan bahasa yang <i>modern</i> , objek-terorientasi, dan memiliki fitur seperti <i>generics</i> , <i>asynchronous programming</i> , dan <i>type inference</i> .
	Flutter	<i>Frontend</i>	Flutter adalah <i>framework open-source</i> yang digunakan untuk mengembangkan aplikasi <i>mobile</i> untuk iOS dan Android, serta aplikasi web dan desktop. Flutter menggunakan bahasa pemrograman Dart sebagai bahasa dasarnya.
	Node.js	<i>Backend</i>	Node.js adalah <i>platform runtime JavaScript</i> yang digunakan untuk menjalankan <i>JavaScript</i> di luar web browser. Node.js menyediakan lingkungan <i>runtime JavaScript</i> yang dapat dijalankan pada sistem operasi seperti Windows, Linux, dan

			Mac. Dengan Node.js, developer dapat menggunakan JavaScript untuk menulis <i>server-side script</i> yang dapat digunakan untuk menjalankan aplikasi web, mengakses <i>database</i> , dan melakukan operasi <i>I/O</i> seperti membaca dan menulis <i>file</i> .
	NestJS	<i>Metaframework</i>	Nest.js adalah framework JavaScript yang dibangun di atas Node.js, yang digunakan untuk membuat aplikasi <i>server-side</i> yang <i>scalable</i> dan <i>performant</i>
	Firebase	<i>Service / Authentication / Push Notifications</i>	<i>Firebase Authentication</i> Layanan ini digunakan untuk <i>mengauthentikasi</i> pengguna aplikasi. <i>Firebase Authentication</i> mendukung berbagai metode otentikasi seperti login dengan email dan password, login dengan akun media sosial seperti Google, Facebook, dan Twitter, dan lainnya. <i>Firebase Authentication</i> juga menyediakan fitur untuk mengelola hak akses pengguna dengan menggunakan <i>Role</i>

			<p><i>Based Access Control (RBAC).</i></p> <p><i>Firebase Cloud Messaging (FCM)</i> Layanan ini digunakan untuk mengirimkan <i>push notifications</i> ke aplikasi mobile. <i>FCM</i> memungkinkan developer untuk mengirimkan pesan ke pengguna aplikasi secara <i>real-time</i> dan menargetkan pesan ke pengguna yang spesifik. <i>FCM</i> juga dapat digunakan untuk mengirimkan pesan ke aplikasi web.</p>
	PostgreSQL	Database	<p><i>PostgreSQL</i> adalah sistem manajemen basis data relasional (RDBMS) yang <i>open source</i> dan dapat digunakan secara gratis. PostgreSQL dikembangkan oleh PostgreSQL Global Development Group dan dapat digunakan pada berbagai sistem operasi seperti Windows, Linux, dan Mac. PostgreSQL sangat fleksibel dan dapat digunakan dalam berbagai jenis proyek, mulai dari aplikasi web hingga aplikasi <i>enterprise</i></p>
	Cloudinary	Storage	<p>Cloudinary adalah layanan <i>cloud-based</i> yang digunakan</p>



			untuk mengelola dan menyajikan media seperti gambar, video, dan audio. Cloudinary memungkinkan <i>developer</i> untuk mengunggah, menyimpan, dan mengelola media dari satu tempat dan menyajikan media dengan performa yang baik
	Play Console	<i>Publishing</i>	Google Play Console adalah platform yang digunakan oleh <i>developer</i> untuk mengelola dan memublikasikan aplikasi mereka di Google Play Store, yang merupakan toko aplikasi resmi untuk perangkat Android.
	Vercel	<i>Publishing</i>	Vercel digunakan untuk <i>deploy</i> aplikasi web secara mudah dan cepat.











### II.3. Alat dan Bahan Penelitian

Untuk mem *build* aplikasi ini penulis menggunakan hardware dan software sebagai berikut.

Nama Part	Keterangan
<i>CPU</i>	Intel Core i7 4700MQ 2.4GHZ
<i>RAM</i>	16 GB
<i>Graphic</i>	NVIDIA Geforce GTX750M
<i>Storage</i>	WDC WDS500G2B0A-00SM50 500.1 GB
Sistem Operasi	<i>Windows 11 Home Single Language 22H2</i>

<i>API Testing</i>	<i>Thunder Client</i>
<i>Text Editor</i>	<i>Visual Studio Code x64</i> dengan plugins: <ul style="list-style-type: none"> <li>• <i>Better Comments</i></li> <li>• <i>Dart</i></li> <li>• <i>Error Gutters</i></li> <li>• <i>Error Lens</i></li> <li>• <i>ESLint</i></li> <li>• <i>Flutter</i></li> <li>• <i>GetX Snippet</i></li> <li>• <i>Github Copilot</i></li> <li>• <i>Github Pull Request and Issues</i></li> <li>• <i>Sort JSON Object</i></li> <li>• <i>Version Lens</i></li> </ul>
<i>Flutter SDK</i>	<i>Flutter SDK</i> dan semua dependensinya: <ul style="list-style-type: none"> <li>• <i>Flutter SDK 3.3.10</i></li> <li>• <i>Dart SDK 2.18.6</i></li> <li>• <i>Microsoft OpenJDK 11.0.15</i></li> <li>• <i>3x Emulator Android Virtual Devices (Pixel Phone) Based on: Android 12.0 (S) Tag/ABI: google_api_playstore/x86_64</i></li> </ul>
<i>Replit</i>	<i>Hosting REST API</i>
<i>Play Console</i>	<i>Publish aplikasi ke Google Play Store</i>
<i>NestJS / Firebase</i>	<i>NodeJS Environment</i> beserta dependensinya: <ul style="list-style-type: none"> <li>• <i>NestJS 9.2.0</i></li> <li>• <i>NodeJS 18.12.1 LTS</i></li> <li>• <i>Firebase Admin SDK (Untuk autentikasi dan Push Notifications di server)</i></li> <li>• <i>Cloudinary (Untuk upload files ke cloud server)</i></li> </ul>

## II.4. Metode Pembuatan Aplikasi

PRIORITY	PROJECT NAME	START DATE	END DATE	PROJECT STATUS	% COMPLETE
 Normal	Project A	05/12/2021	07/12/2021	In progress	<div><div></div></div>
 High	Project B	01/17/2021	05/17/2021	Done	<div><div></div></div>
 High	Project C	07/14/2021	07/14/2021	In progress	<div><div></div></div>
 Low	Project D	05/14/2021	08/19/2021	Cancelled	<div><div></div></div>
 Normal	Project E	07/30/2021	10/30/2021	In progress	<div><div></div></div>
 Normal	Project F	06/22/2021	08/22/2021	In progress	<div><div></div></div>
 Normal	Project G	05/27/2021	09/27/2021	Done	<div><div></div></div>
 Low	Project H	07/21/2021	07/12/2021	In progress	<div><div></div></div>
 Normal	Project G	05/27/2021	09/27/2021	Done	<div><div></div></div>
 Low	Project H	04/12/2021	07/12/2021	In progress	<div><div></div></div>

**Gambar II.2.** Contoh table yang digunakan dalam proses Analisis Kredit Mikro menggunakan metodologi *Agile*.

Dalam pembuatan aplikasi analisis kredit mikro, kami memutuskan untuk menggunakan metode *Agile* sebagai pendekatan pengembangan. Hal ini karena metode *Agile* memungkinkan kami untuk lebih fleksibel dalam menangani perubahan yang mungkin terjadi selama proses pengembangan dan memastikan bahwa aplikasi yang dihasilkan sesuai dengan kebutuhan klien.

Untuk mengimplementasikan metode *Agile*, kami membagi proyek menjadi beberapa *sprint* yang berlangsung selama beberapa minggu. Setiap *sprint* kami fokus pada menyelesaikan beberapa fitur tertentu yang diinginkan. Sebelum memulai *sprint*, kami mengidentifikasi tujuan yang ingin dicapai dan beberapa *user stories* yang harus diselesaikan dalam *sprint* tersebut.

Selama *sprint*, kami menggunakan *Scrum* sebagai framework untuk mengelola proyek. *Scrum* memungkinkan kami untuk mengelola tugas-tugas yang harus diselesaikan dalam *sprint*, mengidentifikasi masalah yang mungkin terjadi, dan mengkoordinasikan kerja tim. Kami juga menggunakan metode *Pair*

*Programming* untuk meningkatkan kualitas kode dan mempercepat proses pengembangan.

Untuk mengelola kode yang dihasilkan, kami menggunakan *Git* sebagai sistem manajemen versi. Kami membuat *branch* kecil untuk setiap fitur yang dikembangkan, kemudian menggabungkan *branch-branch* tersebut ke dalam *branch* utama sebelum di-*deploy* ke *server* produksi. Ini memungkinkan kami untuk mengelola perubahan yang dilakukan pada setiap fitur secara terpisah dan memastikan bahwa perubahan pada satu fitur tidak mempengaruhi fitur lain. Selain itu, kami juga menggunakan *Continuous Integration (CI)* dan *Continuous Deployment (CD)* untuk memastikan bahwa kode yang dihasilkan selalu dalam kondisi stabil dan dapat di-*deploy* ke *server* produksi dengan cepat.

Kami juga menggunakan beberapa teknologi modern dalam proses pengembangan aplikasi ini. Misalnya, kami menggunakan *framework* Node.js untuk *backend*, NestJS sebagai *metaframework* untuk *backend*, dan Flutter untuk membuat aplikasi *mobile*. Dan untuk mengelola data kami menggunakan ElephantSQL sebagai *database*. Dengan menggabungkan metode *Agile*, *Scrum*, *Pair Programming*, *Git*, *CI/CD*, dan teknologi modern, kami dapat memastikan bahwa aplikasi analisis kredit mikro yang dihasilkan sesuai dengan kebutuhan klien dan dapat digunakan dengan baik oleh pengguna.

## **II.5. Meeting dengan PIC (*Person in Charge*)**

Melakukan meeting dengan PIC dapat memberikan banyak manfaat dalam proses pengembangan aplikasi. Pertama, meeting dapat digunakan untuk membahas dan mengevaluasi progress kerja tim, sehingga dapat diidentifikasi masalah yang mungkin terjadi sebelum menjadi masalah besar. Kedua, meeting dapat digunakan untuk mengkoordinasikan tugas-tugas yang harus diselesaikan dalam jangka waktu tertentu. Ketiga, meeting dapat digunakan untuk mengkomunikasikan perubahan yang dilakukan pada aplikasi atau fitur baru yang ditambahkan. Keempat, meeting dapat digunakan untuk mengevaluasi hasil

pengujian aplikasi dan menentukan perbaikan yang perlu dilakukan. Kelima, meeting dapat digunakan untuk mengevaluasi rencana maintenance dan updating aplikasi. Dengan melakukan meeting dengan PIC secara teratur, dapat memastikan bahwa aplikasi yang dihasilkan sesuai dengan kebutuhan klien dan dapat digunakan dengan baik oleh pengguna.

Berikut ini adalah jadwal meeting dengan PIC untuk pengembangan aplikasi analisis kredit mikro:

Meeting	Topik	Keterangan
#1	Analisis Produk	Membahas tentang analisis produk yang akan dikembangkan, termasuk tujuan aplikasi, target pasar, dan kebutuhan pengguna.
#2	Desain Sistem	Membahas desain sistem aplikasi, termasuk arsitektur aplikasi, desain database, dan desain <i>API</i> .
#3	Pembangunan Fitur	Membahas progress pembangunan fitur aplikasi dan mengevaluasi progress kerja tim.
#4	Uji Coba dan Perbaikan	Membahas hasil pengujian aplikasi dan mengevaluasi apakah ada bug atau masalah yang harus diperbaiki.
#5	Testing dan Rekues Fitur	Membahas hasil pengujian aplikasi dan mengevaluasi apakah ada fitur baru yang perlu ditambahkan atau perbaikan yang perlu dilakukan.
#6	Integrasi <i>Third Party Service</i>	Membahas progress integrasi layanan pihak ketiga seperti <i>Firebase Auth</i> , <i>Firebase Cloud Messaging</i> , dan <i>Cloudinary</i> , serta mengevaluasi apakah

		ada masalah yang perlu diperbaiki.
#7	<i>Role Based Access Control</i>	Membahas implementasi Role Based Access Control dan mengevaluasi apakah sistem tersebut berfungsi dengan baik.
#8	<i>Deployment</i>	Membahas proses <i>deployment</i> aplikasi akan digunakan.
#9	Analisis Performa	Membahas hasil analisis performa aplikasi dan mengevaluasi apakah ada perbaikan yang perlu dilakukan.
#10	<i>Maintenance</i> dan <i>Updating</i>	Membahas rencana <i>maintenance</i> dan <i>updating</i> aplikasi, serta mengevaluasi apakah ada fitur baru yang perlu ditambahkan.

## II.6. Job Description (Tugas yang dikerjakan)

Berikut ini adalah deskripsi pekerjaan yang akan ditangani oleh kami, mulai dari analisis produk hingga proses *maintenance* dan *updating* aplikasi, serta mengevaluasi hasil pengujian aplikasi dan perbaikan yang perlu dilakukan. Selain itu, kami juga akan ditugaskan untuk mengkoordinasikan tugas-tugas yang harus diselesaikan dalam jangka waktu tertentu dan mengevaluasi rencana *maintenance* dan *updating* aplikasi.

No	Nama	Job Description
1.	Novian Andika	<ul style="list-style-type: none"> <li>• <i>Full-stack developer</i></li> <li>• Menulis laporan ini</li> <li>• Membuat isi presentasi</li> <li>• Mendesain UI UX secara langsung di Flutter</li> <li>• Membangun <i>REST API</i> untuk disambungkan ke Flutter</li> </ul>

		<p>menggunakan NodeJS</p> <ul style="list-style-type: none"> <li>• Integrasi <i>Firebase</i> ke NodeJS (<i>Authentication</i> dan <i>Cloud Messaging</i> menggunakan <i>Firebase Admin SDK</i>)</li> <li>• Mengonversi formula atau rumus <i>spreadsheet</i> excel ke dalam Dart</li> <li>• Membangun antarmuka aplikasi dengan Flutter mulai dari <i>slicing</i>, <i>state management</i>, <i>unit testing</i>, integrasi <i>third party service</i>, dan <i>fetch / receive</i> data dari <i>REST API</i></li> <li>• Mempublikasikan Aplikasi ke <i>Play Store</i> dengan <i>Play Console</i></li> <li>• Membuat website dokumentasi aplikasi untuk keperluan <i>user guide</i></li> <li>• Membuat website Dashboard Admin untuk keperluan management user</li> <li>• Mendokumentasikan <i>REST API</i> dengan Swagger</li> <li>• Memanage dan <i>Maintenance</i> aplikasi agar berjalan dengan baik.</li> </ul>
2.	Azmi Badhiuz Z	<ul style="list-style-type: none"> <li>• Mendesain <i>layout</i> Powerpoint untuk presentasi</li> <li>• Desain gambar <i>Feature graphic image</i> di Goole Playstore</li> <li>• Desain gambar <i>phone screenshot</i> di Google Playstore</li> <li>• Membuat video highlight aplikasi</li> </ul>

## II.7. Analisis Biaya dan Anggaran

Analisis biaya dan anggaran merupakan bagian yang penting dalam proses pengembangan aplikasi. Sebelum memulai proyek, perlu dilakukan analisis biaya yang dibutuhkan untuk menentukan apakah proyek tersebut layak untuk dilakukan. Hal ini sangat penting untuk memastikan bahwa proyek dapat diselesaikan dalam batas waktu dan anggaran yang telah ditentukan.

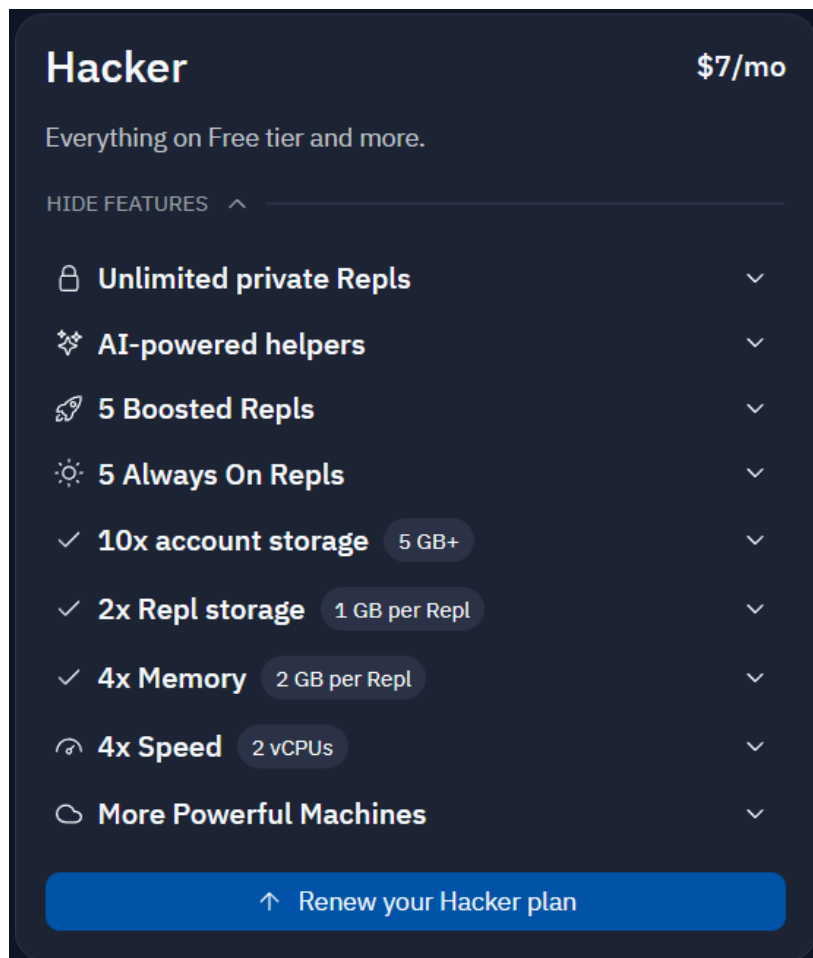
Dalam proses pengembangan aplikasi ini, analisis biaya dan anggaran akan dilakukan untuk menentukan biaya yang dibutuhkan untuk infrastruktur, peralatan, dan sumber daya manusia. Selain itu, akan dilakukan juga analisis *cost-benefit* untuk mengevaluasi kelayakan proyek dan menyesuaikan anggaran jika diperlukan. Teknik *forecasting* dan *budgeting* juga akan digunakan untuk memprediksi dan mengendalikan biaya pengembangan aplikasi.

Berikut adalah biaya yang digunakan untuk pengembangan aplikasi Analisis Kredit Mikro:

Item	Harga	Konversi Rupiah
Replit <i>Hacker Plan</i>	\$7 / Bulan	Rp. 106,239.00
Google Play Console	\$25 Lifetime	Rp. 400,669.00

Selain itu, akan dilakukan pemantauan pengeluaran selama proses pengembangan dan akan dilampirkan *invoice-invoice* yang diterima sebagai bukti pengeluaran. Hal ini dilakukan untuk menjamin transparansi dan akuntabilitas dalam pengelolaan anggaran proyek. Berikut adalah invoice 2 *items* diatas:



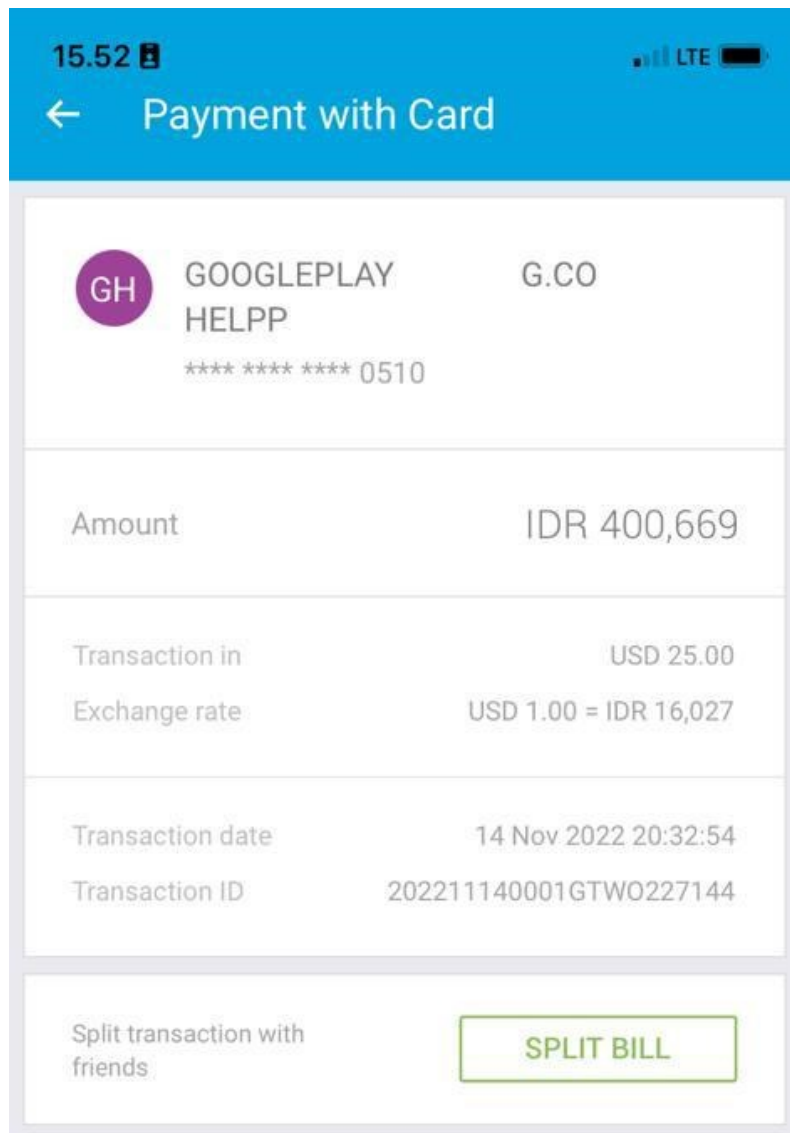


**Gambar II.3.** *Replit Hacker Plan* digunakan untuk mendeploy *backend* yang digunakan sebagai *server*.

Replit menyediakan beberapa pilihan plan berbayar yang disebut "Hacker Plans" yang memberikan fitur-fitur tambahan dan dukungan yang lebih baik dibandingkan dengan plan gratis. Fitur-fitur yang ditawarkan oleh Hacker Plans antara lain:

- Pengguna dapat mengaktifkan fitur *Always On* agar server tidak tertidur saat tidak digunakan dan bisa on sampai 24/7.
- Pengguna dapat menambahkan custom domain untuk repl mereka
- Pengguna dapat menambahkan custom SSL certificate untuk repl mereka
- Pengguna dapat menambahkan fitur-fitur tambahan seperti webhooks, cron jobs, dan banyak lagi

- Pengguna dapat menambahkan kapasitas storage yang lebih besar
- Pengguna dapat mendapatkan dukungan dari tim replit lebih cepat
- Pengguna dapat menambahkan fitur-fitur tambahan lainnya yang dapat membantu dalam pengembangan aplikasi.



**Gambar II.4.** *Invoice Google Play Console. Play Console digunakan untuk publishing aplikasi Android ke Play Store*

*Google Play Console* merupakan platform yang dikembangkan oleh Google untuk mempublikasikan aplikasi android. Keuntungan dari menggunakan *Google Play Console* untuk mempublikasikan aplikasi android adalah akses yang

luas ke pengguna android di seluruh dunia, serta proses publikasi yang mudah dan cepat. Pengguna dapat mengupload aplikasinya, mengelola aplikasi yang sudah di publikasikan, melakukan analisis dan pemantauan aplikasi, serta mengelola fitur-fitur yang tersedia seperti Monetisasi, Analisis, dan Layanan Pembaruan. Selain itu, Google Play Console juga menyediakan fitur-fitur seperti pemantauan *crash*, pengujian *beta*, dan dukungan multi-bahasa yang membantu pengembang dalam mengelola aplikasi mereka dengan lebih baik.

Untuk mempublikasikan aplikasi android dan membutuhkan biaya sebesar \$25 untuk registrasi. Pembayaran ini dilakukan sekali saja dan akan berlaku untuk selamanya. Hal ini dilakukan agar Google dapat menjamin keamanan dan kualitas aplikasi yang dipublikasikan di platform mereka.

Proses registrasi ini digunakan untuk melakukan verifikasi pengembang dan menjamin bahwa aplikasi yang dipublikasikan benar-benar dikembangkan oleh pengembang yang didaftarkan. Hal ini bertujuan untuk mengurangi risiko aplikasi yang tidak berkualitas atau aplikasi yang mengandung malware yang dapat merugikan pengguna. Selain itu, biaya ini juga digunakan untuk menjaga kualitas dari aplikasi yang dipublikasikan, Google dapat menyediakan dukungan dan layanan yang baik untuk pengembang dan pengguna.

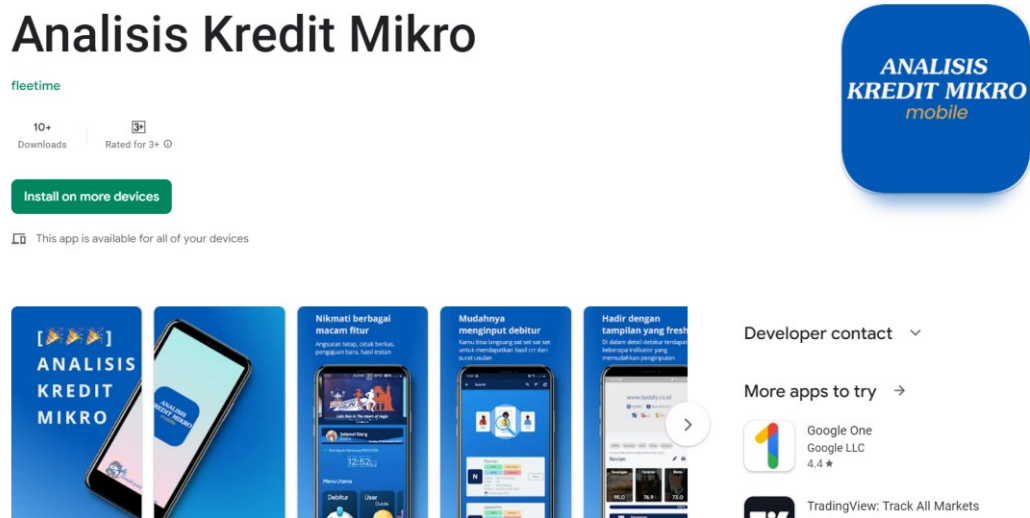
Pembayaran \$25 ini juga digunakan untuk menjaga ekosistem yang sehat dan stabil. *Google Play Console* menyediakan berbagai fitur yang membantu pengembang dalam mengelola aplikasi mereka. Misalnya, fitur analisis yang membantu pengembang dalam menganalisis performa aplikasi, fitur monetisasi yang membantu pengembang dalam menghasilkan uang dari aplikasi dan fitur pengujian beta yang membantu pengembang dalam menguji aplikasinya sebelum dirilis. Selain itu, Google Play Console juga memberikan dukungan yang baik dan cepat sehingga pengembang dapat mengelola aplikasi mereka dengan lebih baik.

## BAB III

### HASIL DAN PEMBAHASAN

#### III.1. Hasil Penelitian

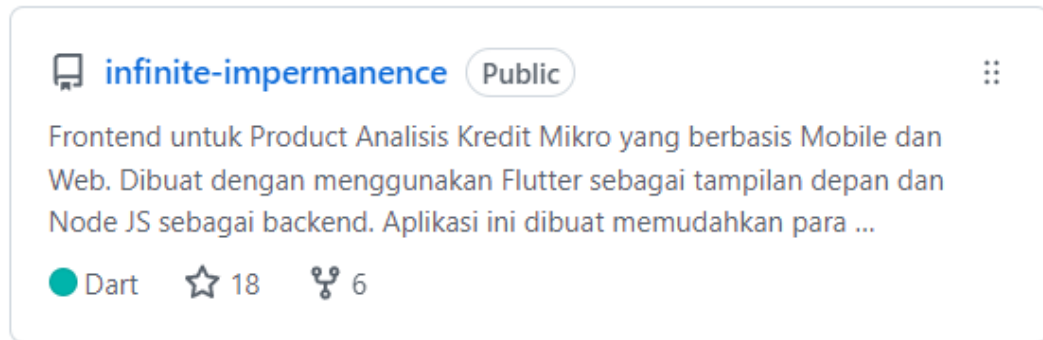
Hasil dari penelitian ini adalah sebuah aplikasi Analisis Kredit Mikro yang dikembangkan menggunakan metode *agile* dan telah di publish di Google Playstore. Selain itu, *source code* dari aplikasi ini juga tersedia di Github, sehingga dapat digunakan oleh pihak lain sebagai referensi atau untuk keperluan pengembangan lebih lanjut. Aplikasi ini dikembangkan dengan menggunakan beberapa teknologi seperti Flutter untuk *front-end*, NodeJS dan NestJS untuk *back-end* dan Firebase untuk autentikasi dan *push notification*. Aplikasi ini sudah di test dan diuji coba oleh PIC sehingga dapat digunakan dengan baik oleh para pemakai.



**Gambar III.1.** Aplikasi Analisis Kredit Mikro yang sudah *listing* di *Google Play Store* dan sudah siap diunduh oleh para *users*.

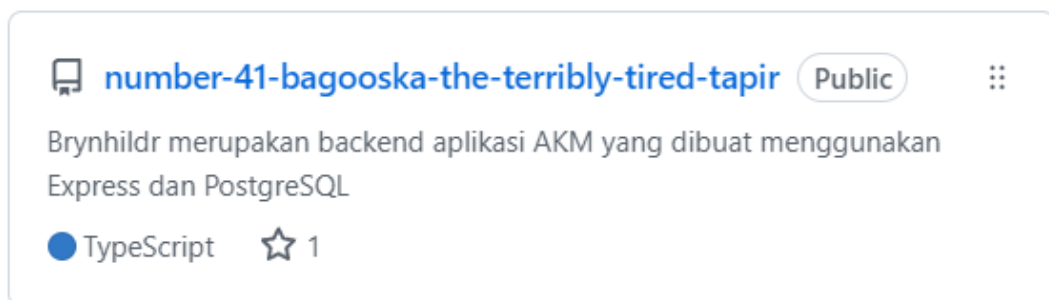
Setelah aplikasi ini di *publish* di *Google Playstore*, ada beberapa biaya yang harus diperhitungkan untuk menjaga agar aplikasi dapat berjalan dengan baik. Salah satu diantaranya adalah biaya *maintenance*, yang diperlukan untuk memperbaiki *bug*, menambah fitur-fitur baru, dan memastikan aplikasi tetap

kompatibel dengan perangkat dan sistem operasi terbaru. Biaya *maintenance* ini harus diperhitungkan secara berkala agar aplikasi tetap dapat digunakan dengan baik oleh para pengguna.



**Gambar III.2.** *Source Code frontend* Flutter yang ditempatkan di *Github*.

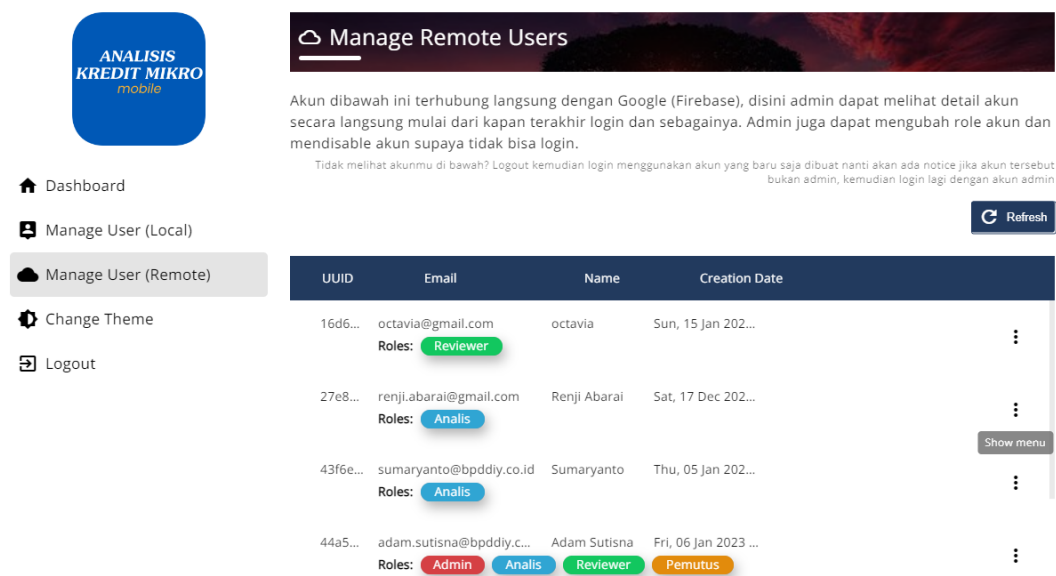
Selain itu, biaya untuk sewa *hosting server* juga harus diperhitungkan. Aplikasi ini menggunakan server untuk menjalankan *REST API* yang digunakan oleh aplikasi mobile dan juga untuk menyimpan data-data user. Oleh karena itu, biaya untuk menyewa server yang cukup *powerful* dan stabil harus diperhitungkan agar aplikasi dapat berjalan dengan baik.



**Gambar III.3.** *Source Code backend* NodeJS yang ditempatkan di *Github*.

Biaya lainnya yang harus diperhitungkan adalah biaya untuk integrasi dengan third party service seperti Firebase Auth, Firebase Cloud Messaging, dan Cloudinary yang digunakan untuk autentikasi dan *push notification*. Biaya untuk integrasi ini harus diperhitungkan untuk menjaga agar aplikasi dapat berjalan dengan baik dan dapat digunakan oleh para pengguna dengan baik.

Selain aplikasi mobile yang digunakan oleh para pengguna, dalam penelitian ini juga dikembangkan sebuah aplikasi *dashboard* admin. Aplikasi ini digunakan untuk mengelola *user* yang akan *login* ke aplikasi, seperti memberikan *role* dan hak akses, menambahkan atau menghapus user, dan mengecek data user yang telah di input. Aplikasi dashboard admin dikembangkan menggunakan *Flutter Web*, sehingga dapat diakses melalui browser dan dapat digunakan oleh admin dari mana saja.

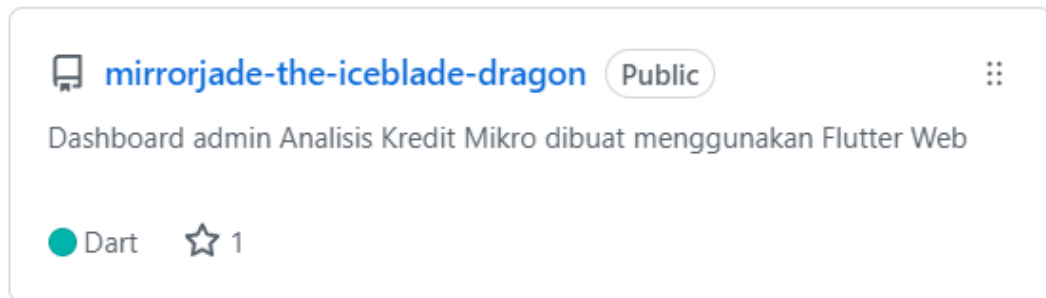


**Gambar III.4.** *Dashboard* admin yang sudah di *deploy*.

Aplikasi dashboard admin ini sangat penting untuk menjaga keamanan dan privasi data user yang telah diinput ke aplikasi. Dengan adanya aplikasi dashboard admin, admin dapat dengan mudah mengelola dan mengontrol user yang akan login ke aplikasi serta menjaga agar data user tidak diakses oleh orang yang tidak berwenang. Selain itu, aplikasi dashboard admin ini juga memudahkan admin dalam mengelola user dan mengambil tindakan jika terjadi masalah dengan user yang telah diinput.

Kemudian *dashboard* admin yang dikembangkan dalam penelitian ini di *deploy* menggunakan Vercel, sebuah platform untuk deployment web yang memungkinkan *developer* untuk mengimport aplikasi dari Github dan secara

otomatis di-*deploy* di web tanpa perlu melakukan konfigurasi apapun. Vercel menawarkan fitur yang memudahkan developer untuk melakukan deployment aplikasi web, seperti *automatic HTTPS*, *global CDN*, dan *continuous deployment*. Dengan menggunakan Vercel, developer dapat menghemat waktu dan tenaga dalam melakukan konfigurasi server dan fokus pada pengembangan aplikasi.



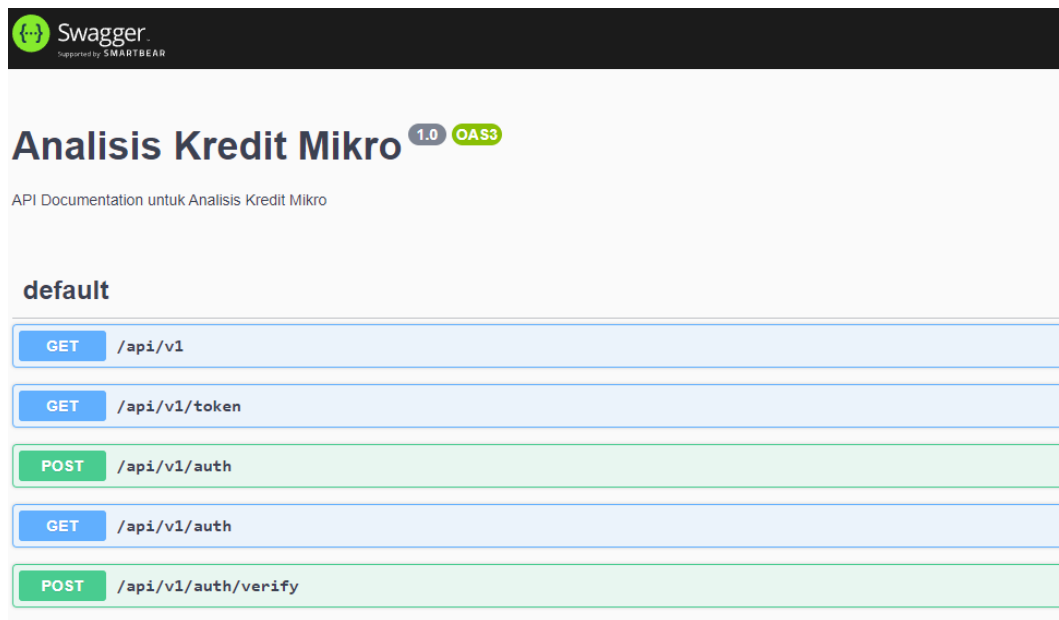
**Gambar III.5.** *Source Code Dashboard* admin yang tersedia di Github.

Dalam penelitian ini, aplikasi yang dikembangkan dilengkapi dengan dokumentasi *API* yang dibuat menggunakan Swagger. Swagger merupakan *tool* yang digunakan untuk meng-*generate* dokumentasi *API* secara otomatis, sehingga *developer* dapat dengan mudah memahami bagaimana *API* tersebut digunakan. Dengan dokumentasi *API* yang lengkap, *developer* dapat dengan mudah memahami bagaimana aplikasi ini bekerja dan membuat integrasi dengan aplikasi lain.

Selain dokumentasi *API*, aplikasi ini juga dilengkapi dengan *user guide* yang dibuat menggunakan Docsify. Docsify adalah sebuah library yang digunakan untuk membuat dokumentasi *online* dengan cepat dan mudah. Dengan Docsify, *developer* dapat dengan mudah membuat *user guide* yang sederhana namun informatif, sehingga pengguna dapat dengan mudah memahami cara menggunakan aplikasi ini.

*User guide* yang dibuat menggunakan Docsify juga dapat diakses secara online, sehingga pengguna dapat dengan mudah menemukannya dan mempelajarinya kapan saja. Dengan adanya *user guide* yang lengkap dan mudah

diakses, pengguna dapat dengan mudah memahami cara menggunakan aplikasi ini dan memanfaatkannya dengan baik.



**Gambar III.6.** Dokumentasi *REST API* menggunakan Swagger.

Kedua dokumentasi API dan user guide yang dibuat menggunakan Swagger dan Docsify ini akan sangat membantu developer dalam mengembangkan aplikasi ini dan pengguna dalam menggunakan aplikasi ini, sehingga aplikasi ini akan lebih mudah digunakan dan digunakan secara optimal.



**Gambar III.7.** Dokumentasi *User Guide* menggunakan Docsify



Berikut ini adalah beberapa tautan hasil penelitian proyek diatas:

<b>Nama</b>	<b>Type</b>	<b>Tautan</b>
Analisis Kredit Mikro ( <i>Product</i> )	<i>Mobile</i> / Play Store	<a href="https://s.id/akm-app">https://s.id/akm-app</a>
Analisis Kredit Mikro ( <i>Dashboard Admin</i> )	Website	<a href="https://s.id/akm-das">https://s.id/akm-das</a>
Analisis Kredit Mikro ( <i>REST API</i> )	<i>Webservice</i> / Dokumentasi	<a href="https://s.id/akm-api">https://s.id/akm-api</a>
Analisis Kredit Mikro ( <i>Frontend Code</i> )	<i>Frontend</i> / <i>Source</i> <i>Code</i> / Flutter	<a href="https://s.id/akm-flu">https://s.id/akm-flu</a>
Analisis Kredit Mikro ( <i>Backend Code</i> )	<i>Backend</i> / <i>Source</i> <i>Code</i> / NodeJS	<a href="https://s.id/akm-nod">https://s.id/akm-nod</a>
Analisis Kredit Mikro ( <i>Dashboard Code</i> )	<i>Frontend</i> / <i>Souce</i> <i>Code</i> / Flutter	<a href="https://s.id/akm-adm">https://s.id/akm-adm</a>
Analisis Kredit Mikro ( <i>User Guide</i> )	<i>Tutorial</i> / <i>User</i> <i>Guide</i> / Website	<a href="https://s.id/akm-tut">https://s.id/akm-tut</a>

### III.2. Persiapan Menjalankan Aplikasi Mobile

Aplikasi *mobile* Analisa Kredit Mikro hanya dapat digunakan pada perangkat yang menjalankan sistem operasi Android. Hal ini dikarenakan saat pengembangan aplikasi menggunakan Flutter, penulis menggunakan sistem operasi Windows sebagai platform pengembangan. Sistem operasi Windows hanya memungkinkan untuk melakukan build aplikasi pada *platform* Android, sehingga aplikasi ini tidak dapat digunakan pada perangkat yang menjalankan sistem operasi lain seperti *iOS*. Namun, pengembangan aplikasi pada *platform* lain dapat dipertimbangkan pada masa yang akan datang, untuk meningkatkan aksesibilitas dan menjangkau lebih banyak pengguna.

Aplikasi *mobile* Analisa Kredit Mikro memiliki beberapa minimal sistem *requirement* yang harus dipenuhi agar dapat dijalankan dengan baik pada perangkat android. Beberapa diantaranya adalah minimal versi sistem operasi

android yang didukung, kapasitas memori yang dibutuhkan, serta spesifikasi hardware lainnya seperti prosesor dan GPU. Pengguna harus memastikan bahwa perangkat yang digunakan memenuhi minimal sistem requirement tersebut agar dapat menjalankan aplikasi dengan lancar dan tanpa kendala.

Berikut adalah sistem minimal dan sistem yang direkomendasikan oleh *developer*:

Komponen	Minimum	Rekomendasi
OS	Android 6	Android 10+
RAM	4 GB	32 GB
Storage	20MB	100MB <i>Free Space</i>

Selain minimal sistem *requirement* yang harus dipenuhi, Aplikasi *mobile* Analisa Kredit Mikro juga memerlukan beberapa *permission* yang harus diaktifkan agar dapat berjalan dengan baik. Beberapa *permission* yang harus diaktifkan antara lain akses ke kamera, akses ke storage, dan akses ke lokasi. Pengguna harus mengaktifkan *permission-permission* tersebut agar aplikasi dapat bekerja dengan sempurna dan menyediakan fitur yang sesuai dengan yang diharapkan. Tanpa *permission* yang diaktifkan, beberapa fitur dari aplikasi mungkin tidak dapat diakses dan digunakan oleh pengguna

Berikut adalah beberapa *permission* dan penjelasan kenapa diaktifkan:

Nama	Permission yang dibutuhkan	Keterangan
<i>Location</i>	<ul style="list-style-type: none"> <li>• <i>Access precise location only on foreground</i></li> <li>• <i>Access appropriate location (network-based) only on foreground</i></li> </ul>	<i>Permission</i> ini digunakan untuk melacak lokasi analisis saat menginput agunan tanah dan bangunan, sehingga posisi terdeteksi secara otomatis menggunakan <i>GPS</i> . Fitur ini hanya berjalan saat aplikasi

		dibuka dan tidak berjalan di latar belakang, sehingga tidak merugikan daya baterai.
<i>Storage</i>	<ul style="list-style-type: none"> <li>• <i>read the contents of your USB storage.</i></li> <li>• <i>modify or delete the contents of your USB storage</i></li> </ul>	<i>Permission</i> ini digunakan untuk membaca file dari penyimpanan perangkat untuk keperluan <i>upload</i> dokumen atau gambar. Dan mengakses <i>storage</i> untuk mendownload photo atau file dari aplikasi
<i>Photos/Media/Files</i>	<ul style="list-style-type: none"> <li>• <i>read the contents of your USB storage.</i></li> <li>• <i>modify or delete the contents of your USB storage</i></li> </ul>	<i>Permission</i> ini digunakan untuk mengakses kamera perangkat dan memungkinkan pengguna untuk mengambil foto. Dengan <i>permission</i> ini, pengguna dapat mengambil foto dengan mudah dan cepat melalui aplikasi yang menggunakan <i>permission</i> ini.
<i>Other</i>	<ul style="list-style-type: none"> <li>• <i>receive data from Internet.</i></li> <li>• <i>view network connections</i></li> <li>• <i>full network access</i></li> <li>• <i>run at startup.</i></li> <li>• <i>control vibration</i></li> <li>• <i>prevent device from sleeping</i></li> </ul>	<i>Permission</i> ini digunakan untuk mengakses jaringan internet dan memungkinkan aplikasi untuk terhubung dengan <i>backend server</i> . Dengan <i>permission</i> ini, aplikasi dapat mengirim dan menerima data dari <i>server</i> , sehingga dapat menyediakan fitur-fitur yang diperlukan oleh pengguna. Fitur-fitur tersebut seperti, mengambil data dari <i>server</i> , mengirim data ke <i>server</i> , <i>login</i> . Selain itu,

		<p><i>permission</i> ini juga memungkinkan aplikasi untuk menerima <i>Firebase Cloud Messaging</i> atau <i>push notification</i>. <i>Push notification</i> merupakan fitur yang sangat berguna untuk aplikasi, karena dapat digunakan untuk memberikan informasi penting kepada pengguna, seperti pemberitahuan, <i>update</i>, atau promosi. Ini membuat aplikasi lebih interaktif dan meningkatkan pengalaman pengguna.</p>
--	--	---

### III.3. Persiapan Menjalankan Aplikasi Web Dashboard

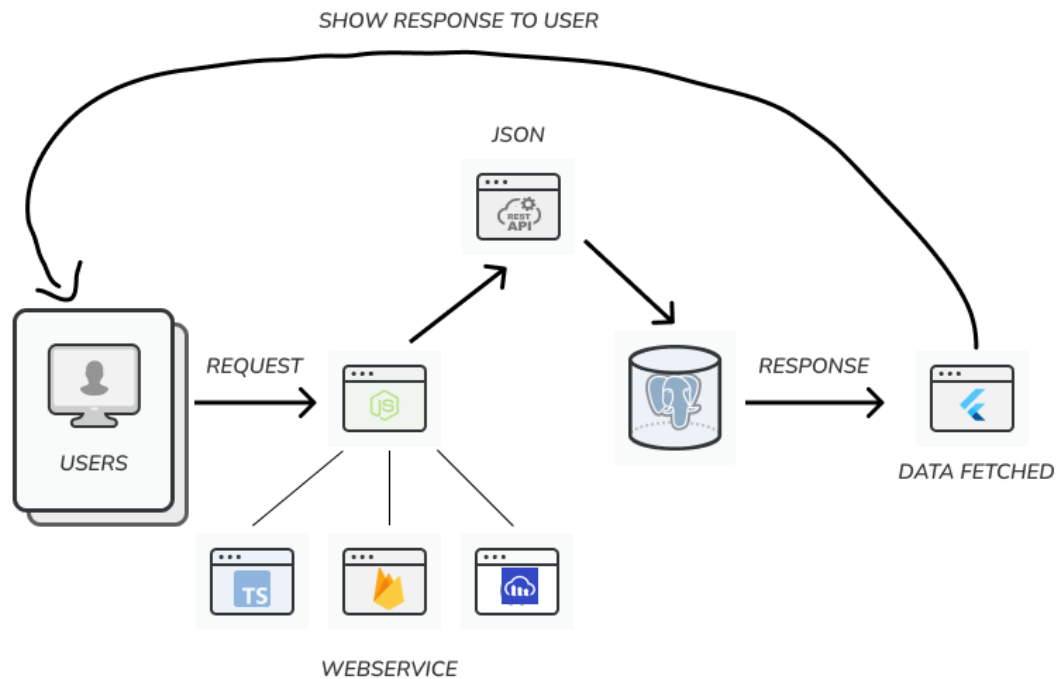
Untuk menjalankan aplikasi admin dashboard, beberapa persiapan yang perlu dilakukan antara lain:

1. Pastikan koneksi internet stabil, karena aplikasi ini memerlukan akses internet untuk terhubung dengan backend server.
2. Pastikan anda telah login dengan akun yang memiliki hak akses sebagai admin
3. Pastikan sudah terinstall web browser versi terbaru (Chrome, Firefox, Edge)
4. Kunjungi link dashboard administrator yang telah disediakan

### III.4. Pembahasan

Aplikasi Analisa Kredit Mikro ini dibangun dengan menggunakan teknologi *mobile development* yang saat ini sedang populer, yaitu Flutter. Saat pengguna membuka aplikasi dan melakukan login, sistem akan mengecek kebenaran data yang diinputkan dengan mengambil data dari *Firebase*

*Authentication.* Setelah berhasil login, pengguna dapat mengakses fitur-fitur yang tersedia seperti input data debitur baru, melihat data debitur yang sudah diinputkan sebelumnya, dan melakukan pencarian data debitur.



**Gambar III.8.** Contoh alur jalannya aplikasi Analisa Kredit Mikro secara simpel

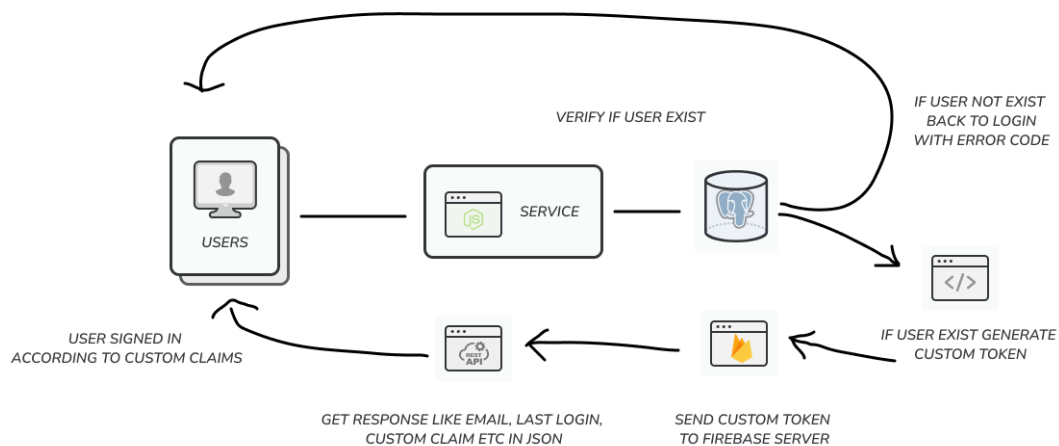
Proses input data debitur baru dilakukan dengan mengirimkan data yang diinputkan ke server melalui *REST API* yang dibangun menggunakan NodeJS. Server kemudian akan menyimpan data tersebut ke dalam database yang digunakan yaitu PostgreSQL yang dihosting di *ElephantSQL*. Kemudian server akan mengirimkan *response* berupa konfirmasi data berhasil disimpan ke aplikasi *mobile*.

Fitur pencarian data debitur dilakukan dengan mengirimkan *request* ke *server* yang kemudian akan mengambil data dari *database* dan mengirimkan *response* berupa data debitur yang sesuai dengan kriteria pencarian. Data yang ditampilkan di aplikasi *mobile* merupakan hasil dari proses *filtering* dan *sorting* yang dilakukan oleh *server*.

Selain itu, aplikasi ini juga dilengkapi dengan fitur-fitur lain seperti perhitungan angsuran, laporan keuangan, dan integrasi dengan *third party service* seperti *Firebase Cloud Messaging* untuk *push notification* dan *Clouinary* untuk penyimpanan berkas. Dengan adanya fitur-fitur tersebut diharapkan dapat membantu para pemakai dalam menganalisis kredit mikro dan membuat keputusan yang tepat.

Penjelasan di atas hanyalah sekilas tentang cara kerja aplikasi. Untuk lebih detail tentang cara kerja aplikasi akan dijabarkan dalam sub-bab tersendiri, mulai dari peran user, pengaturan hak akses (*RBAC*), *push notification*, hingga proses pengajuan.

#### III.4.1. Pembahasan #1: Autentikasi



**Gambar III.9.** Alur autentikasi aplikasi Analisis Kredit Mikro

Autentikasi aplikasi analisis kredit mikro dilakukan dengan menggunakan teknologi *Firebase Admin SDK* yang memungkinkan untuk mengimplementasikan *custom auth* di server. Data user yang diperlukan dalam proses autentikasi sudah tersimpan dalam *database* terlebih dahulu, namun untuk menghubungkan data tersebut dengan aplikasi *mobile*, digunakan pihak ketiga yaitu *Firebase* dengan metode membuat *custom token*. *Custom token* ini seperti *JWT (JSON Web Token)* tapi dibuat oleh *Firebase* sendiri. Metode ini digunakan

untuk memudahkan proses integrasi dengan *Firebase Cloud Messaging (FCM)* untuk keperluan notifikasi saat menerima pengajuan dari *user*.

A screenshot of a code editor window titled 'custom-token.ts'. The code is written in TypeScript and shows the process of creating a custom token using the Firebase Auth SDK. It starts with a constant 'uid' set to 'some-uid'. Then, it calls 'getAuth()' followed by '.createCustomToken(uid)'. A '.then()' callback is provided, which logs the 'customToken' to the console and returns it. A '.catch()' callback is also provided to handle any errors, logging them to the console. The code is as follows:

```
const uid = 'some-uid';

getAuth()
  .createCustomToken(uid)
  .then((customToken) => {
    // Send token back to client
  })
  .catch((error) => {
    console.log('Error creating custom token:', error);
  });
```

**Gambar III.9.** Contoh implementasi pembuatan *customToken* di server

Selain itu, *custom token* juga digunakan untuk mengatur hak akses user atau yang disebut Role Based Access Control (RBAC). RBAC ini diimplementasikan dengan menggunakan *Custom Claims* yang disediakan oleh *Firebase Auth*. *Custom Claims* ini digunakan untuk memberikan role kepada user dan mengatur hak akses user. Selain itu, *custom token* juga digunakan untuk mengatur *session user*, sehingga user dapat tetap login walau dari perangkat yang berbeda.

Pada proses *autentikasi*, user akan melewati proses validasi dan otorisasi. Validasi dilakukan dengan mengecek email dan *password* yang diinputkan oleh *user*, sedangkan otorisasi dilakukan dengan mengecek *role* dan hak akses user. Proses ini dilakukan secara *real-time* dengan mengambil data dari *Firebase Auth* dan mengeceknya dengan data yang ada di server. Jika data cocok, maka user dapat login dan mengakses fitur yang sesuai dengan role dan hak akses yang diberikan oleh admin.

```

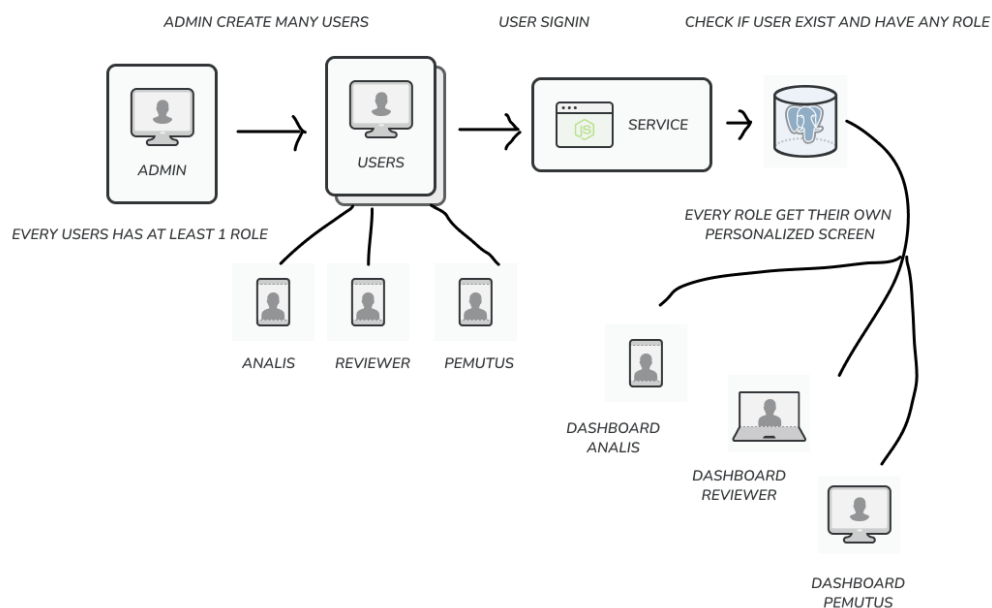
signin.dart

try {
  final userCredential =
    await FirebaseAuth.instance.signInWithCustomToken(token);
  print("Sign-in successful.");
} on FirebaseAuthException catch (e) {
  switch (e.code) {
    case "invalid-custom-token":
      print("The supplied token is not a Firebase custom auth token.");
      break;
    case "custom-token-mismatch":
      print("The supplied token is for a different Firebase project.");
      break;
    default:
      print("Unkown error.");
  }
}

```

**Gambar III.10.** Proses sign in melalui *customToken* yang dibuat tadi.

### III.4.2. Pembahasan #2: *Role-based access control (RBAC)*



**Gambar III.10.** Skema simple implementasi *RBAC* di Analisis Kredit Mikro.



Aplikasi analisis kredit mikro menggunakan sistem *role-based access control* (RBAC) untuk mengatur hak akses user. Terdapat 4 macam role yang diterapkan dalam aplikasi ini, yaitu Analis, *Reviewer*, Pemutus, dan Admin.

1. Analis bertugas untuk menginput data debitur dan melampirkan dokumen pendukung seperti foto-foto agunan dan berkas-berkas yang terkait dengan debitur. Analis juga harus menuliskan analisis yang mendalam mengenai kondisi debitur sebelum mengajukannya untuk ditinjau oleh *Reviewer*.
2. *Reviewer* bertugas untuk meninjau data yang diajukan oleh Analis dan memberikan *feedback* atau rekomendasi yang diperlukan. *Reviewer* juga harus menambahkan bahasan kritis mengenai analisis debitur sebelum mengajukannya ke Pemutus.
3. Pemutus bertugas untuk memutuskan apakah debitur layak untuk diajukan atau tidak. Pemutus harus menganalisis data yang diterima dari *Reviewer* dan memberikan keputusan yang obyektif dan beralasan. Pemutus juga harus menambahkan bahasan mengenai keputusannya.
4. Admin bertugas untuk mengelola user dan hak akses user. Admin harus mengatur hak akses user sesuai dengan role yang ditentukan dan mengawasi aktivitas dari user dalam aplikasi.

*Custom claims* dalam *Firebase Auth* digunakan untuk menambahkan level hak akses pada setiap user yang terdaftar di aplikasi. Dalam hal ini, aplikasi analisis kredit mikro menggunakan *custom claims* untuk mengatur *role* masing-masing user.

Proses implementasi *custom claims* dimulai dengan menambahkan role pada saat user baru login ke dalam aplikasi. *Role* ini kemudian akan disimpan pada *custom claims* yang terdapat di dalam *JWT (JSON Web Token)* yang diterima oleh *client* setelah proses autentikasi berhasil.

Integrasi dengan *PostgreSQL* dilakukan dengan cara menyimpan *role* yang diterima dari *custom claims* ke dalam tabel user di dalam *database*. Sehingga,

setiap kali user melakukan *request*, *role* yang tersimpan dalam *JWT* dapat dengan mudah divalidasi dengan data yang tersimpan di dalam *database*.

Secara keseluruhan, *custom claims* bersama dengan integrasi ke *PostgreSQL* memungkinkan aplikasi untuk memberikan hak akses yang sesuai dengan *role* masing-masing *user* dan menjamin keamanan aplikasi dari akses yang tidak sah.

#### **III.4.3. Pembahasan #3: Struktur Database**

Untuk struktur database *PostgreSQL* pada aplikasi Analisis Kredit Mikro. Dalam hal ini, setiap user analis memiliki banyak data debitur yang terhubung dengan relasi *one to many*. Kemudian, Analis dapat mengisi data debitur dengan berbagai parameter seperti keuangan, karakter, jenis usaha, agunan, dan bisnis. Dari kelima parameter tersebut, akan dihasilkan sebuah nilai CRR atau (*Credit Risk Rating*) yang nantinya akan dinilai dalam proses pengajuan.

Untuk proses pengajuan sendiri, relasinya menggunakan *many to many* dengan user. Hal ini dilakukan agar pengajuan dapat tersinkronisasi dengan *reviewer* dan pemutus, sehingga data tetap sama terintegrasi dan dapat berjalan dengan baik.

Selain itu, untuk menjamin keamanan data, seluruh akses pada database dibatasi oleh sistem autentikasi yang diterapkan. Hanya user yang telah terautentikasi saja yang dapat mengakses dan mengelola data yang tersimpan dalam database.

#### **III.4.4. Pembahasan #4: Push Notification**

Untuk fitur *push notification* Aplikasi Analisis Kredit Mikro menggunakan layanan *Firebase Cloud Messaging (FCM)* dari *Firebase*. Layanan ini memungkinkan untuk mengirim pesan singkat notifikasi kepada user yang menginstall aplikasi. Namun, untuk mengcustom behavior aplikasi agar hanya mengirim notifikasi kepada user yang berhak saja (seperti Analis mengirim pengajuan ke *Reviewer*, *Reviewer* akan mendapat notifikasi dari Analis jika ada pengajuan baru), dibutuhkan implementasi FCM di *server environment* sendiri.

Untuk strategi yang digunakan, dibuatkan tabel *fcmToken* dalam tabel *user* pada basis data kami. *fcmToken* didapatkan ketika pengguna membuka aplikasi. Ketika pengguna belum login, permintaan post untuk mengirimkan *fcmToken* ke server tidak diintersep. Namun, ketika pengguna sudah login, permintaan post akan diintersep untuk mengirim *fcmToken* ke server. Dengan demikian, setiap pengguna akan memiliki *fcmToken* yang unik.

A screenshot of a code editor window titled 'fcmToken.ts'. The code is written in TypeScript and demonstrates how to send an FCM message. It starts with a comment explaining that the registration token comes from the client's FCM SDKs. A constant 'registrationToken' is assigned the value 'YOUR\_REGISTRATION\_TOKEN'. Then, a 'message' object is created with 'data' (containing 'score' and 'time') and 'token' (set to 'registrationToken'). The code then uses 'getMessaging().send(message)' to send the message. It includes a '.then()' callback for successful sending, logging the message ID, and a '.catch()' callback for errors, logging the error details.

```
fcmToken.ts

// This registration token comes from the client
// FCM SDKs.
const registrationToken =
  'YOUR_REGISTRATION_TOKEN';

const message = {
  data: {
    score: '850',
    time: '2:45'
  },
  token: registrationToken
};

// Send a message to the device corresponding to
// the provided
// registration token.
getMessaging().send(message)
  .then((response) => {
    // Response is a message ID string.
    console.log('Successfully sent message:',
response);
  })
  .catch((error) => {
    console.log('Error sending message:', error);
  });
```

**Gambar 5.1.** Contoh *build fcm send request* yang diimplementasikan di server.

Saat mengirim pesan notifikasi, akan dilakukan filter data user dengan mengecek *fcmToken* yang tersimpan di dalam tabel *user*, sehingga hanya *user* yang memiliki *fcmToken* yang cocok dengan yang ditentukan pada saat

pengiriman pesan yang akan menerima notifikasi tersebut. Dengan begitu, dapat mengirim pesan notifikasi yang terkustomisasi dan sesuai dengan kebutuhan user, sehingga proses pengiriman pesan notifikasi lebih efektif dan efisien.

#### **III.4.5. Pembahasan #5: Upload Files dan Foto**

Untuk media pengunggah berkas dan foto kami menggunakan layanan *pihak ketiga*, yaitu Cloudinary. Layanan ini memungkinkan untuk mengupload berkas file seperti gambar dan dokumen ke *cloud server* tanpa mengganggu kinerja dari server yang digunakan. Ini dikarenakan jika *storage* media digabungkan dengan *API server*, maka akan cepat terisi penuh dan menurunkan performa sistem. Dengan menggunakan Cloudinary, dapat menjamin kapasitas storage media yang cukup dan performa yang stabil untuk aplikasi Analisis Kredit Mikro.

Strategi yang digunakan adalah setiap *user* memiliki *relasi one to many* dengan tabel *upload*. Sehingga setiap user dapat mengupload banyak gambar dan files yang terkait dengan data debitur yang input. Hal ini memudahkan proses analisis kredit mikro karena seluruh data dan berkas yang dibutuhkan tersedia dalam satu tempat.

Selain itu, layanan Cloudinary juga menyediakan fitur-fitur seperti pengaturan hak akses, enkripsi data dan backup secara otomatis yang memastikan keamanan data dan berkas yang diupload. Dengan menggunakan layanan ini, kami dapat meningkatkan efisiensi dan efektivitas dalam proses analisis kredit mikro di Bank BPD DIY.

#### **III.5. Fitur Fitur Aplikasi Analisis Kredit Mikro**

1. *Data-driven Credit Risk Rating (CRR) Analysis*: Aplikasi ini menggunakan teknologi analitik canggih untuk mengevaluasi risiko kredit debitur berdasarkan data yang dikumpulkan dari beberapa parameter seperti keuangan, karakter, jenis usaha, agunan, dan bisnis.
2. *Role-based Access Control (RBAC) System*: Aplikasi ini menyediakan sistem kontrol akses berbasis peran yang memungkinkan pengguna untuk

memiliki hak akses yang sesuai dengan perannya masing-masing seperti Analis, Reviewer, Pemutus, dan Admin.

3. *Cloud-based File Upload & Management*: Aplikasi ini menggunakan *cloud server* sebagai media penyimpanan file dan gambar yang diunggah oleh pengguna sehingga memungkinkan untuk mengakses file dan gambar dengan mudah tanpa mengganggu server utama.
4. *Real-time Notification & Synchronization*: Aplikasi ini menyediakan fitur notifikasi *real-time* yang memungkinkan pengguna untuk menerima notifikasi saat ada pengajuan baru atau perubahan data yang dilakukan oleh pengguna lain. Fitur ini juga menyediakan sistem sinkronisasi data yang memastikan data yang ditampilkan sama di setiap perangkat yang digunakan.
5. *Advanced Search & Filtering*: Aplikasi ini menyediakan fitur pencarian dan penyaringan yang canggih yang memungkinkan pengguna untuk mencari dan menyaring data debitur dengan mudah berdasarkan kriteria tertentu seperti nama, range plafond, jenis usaha, dll.
6. *Customizable Reports & Analytics*: Aplikasi ini menyediakan fitur laporan dan analitik yang dapat disesuaikan sesuai kebutuhan pengguna. Serta menyediakan fitur untuk mencetak laporan dari semua yang sudah diinputkan kedalam format pdf.
7. *Automatic Workflow & Approval Process*: Fitur ini memungkinkan proses analisis kredit berjalan secara otomatis sesuai dengan workflow yang telah ditentukan. Mulai dari input data debitur, tinjauan oleh reviewer hingga proses pemutusan yang dilakukan oleh pihak yang berwenang.
8. *Data Security & Compliance*: Aplikasi ini menjamin keamanan dan kerahasiaan data yang diinputkan dengan mengimplementasikan enkripsi dan autentikasi yang kuat serta memenuhi regulasi dan standar keamanan data yang berlaku.

## BAB IV

### KESIMPULAN

#### IV.1. Kesimpulan

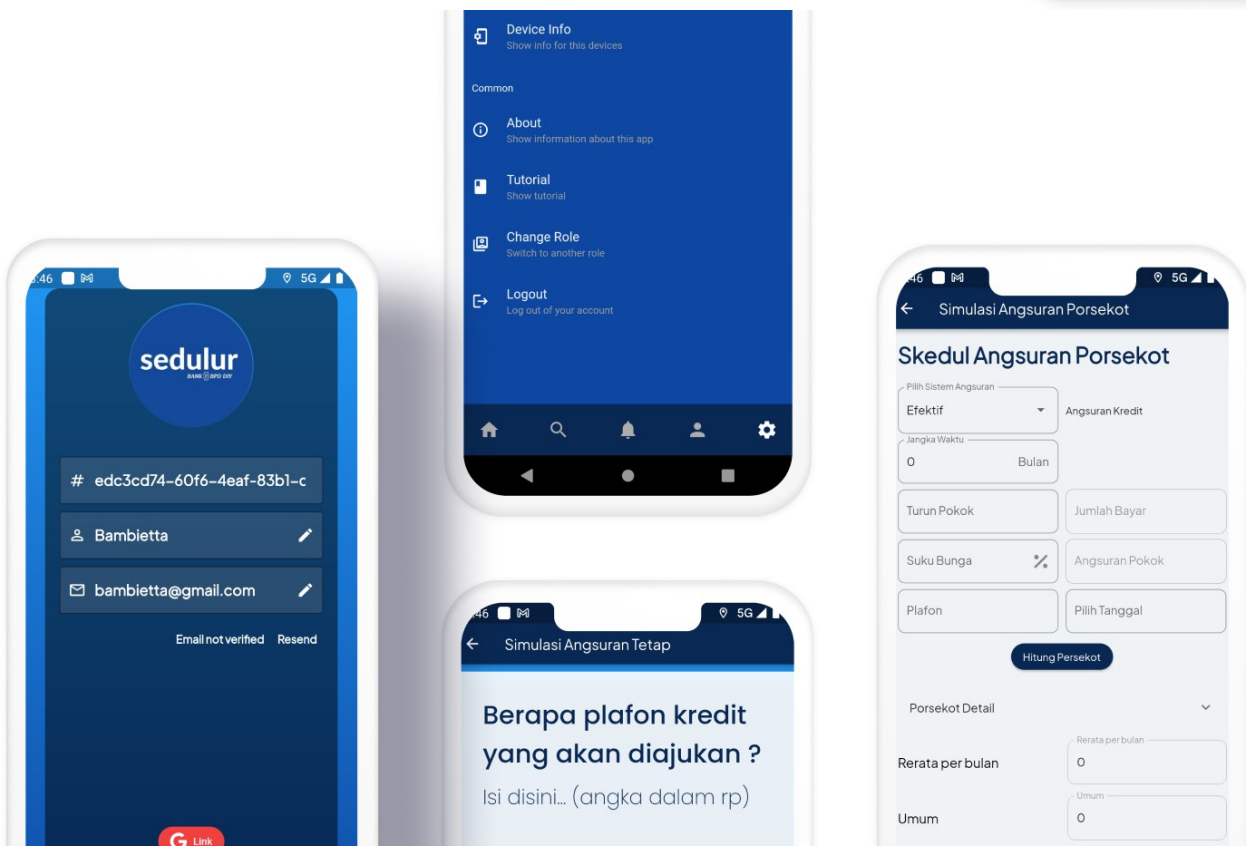
1. Penelitian ini menyoroti penerapan analisis kredit mikro untuk meningkatkan akses kredit bagi usaha kecil, yang diimplementasikan dalam bentuk aplikasi *mobile* dan admin dashboard.
2. Aplikasi dikembangkan menggunakan teknologi Flutter dan dibangun di atas *platform* Android.
3. Penelitian ini menunjukkan bahwa dengan menggunakan analisis kredit mikro, dapat meningkatkan efisiensi dalam proses analisis dan pengambilan keputusan kredit.
4. Implementasi *custom auth* di server Firebase Admin SDK digunakan untuk mengamankan aplikasi dan mengimplementasikan *role-based access control (RBAC)*.
5. Penelitian ini menunjukkan bahwa dengan menggunakan *Firebase Cloud Messaging*, dapat memberikan notifikasi *real-time* kepada user.
6. Aplikasi ini dilengkapi dengan *dokumentasi API* dan *user guide* yang dikembangkan menggunakan Swagger dan Docsify.
7. Penelitian ini menyoroti pentingnya biaya *maintenance* dan sewa *hosting* server dalam pengembangan aplikasi.
8. Hasil dari penelitian ini adalah sebuah aplikasi yang telah dipublikasikan di Google Play Store dan sumber kodenya tersedia di GitHub.
9. Penelitian ini menunjukkan bahwa dengan menggunakan integrasi PostgreSQL, dapat memberikan fleksibilitas dan keamanan dalam pengelolaan data.
10. Secara keseluruhan, penelitian ini menunjukkan bahwa analisis kredit mikro dapat meningkatkan akses kredit bagi usaha kecil dan dapat diimplementasikan dalam bentuk aplikasi mobile yang efisien dan aman.

DAEV

## **LAMPIRAN**

## LAMPIRAN #1

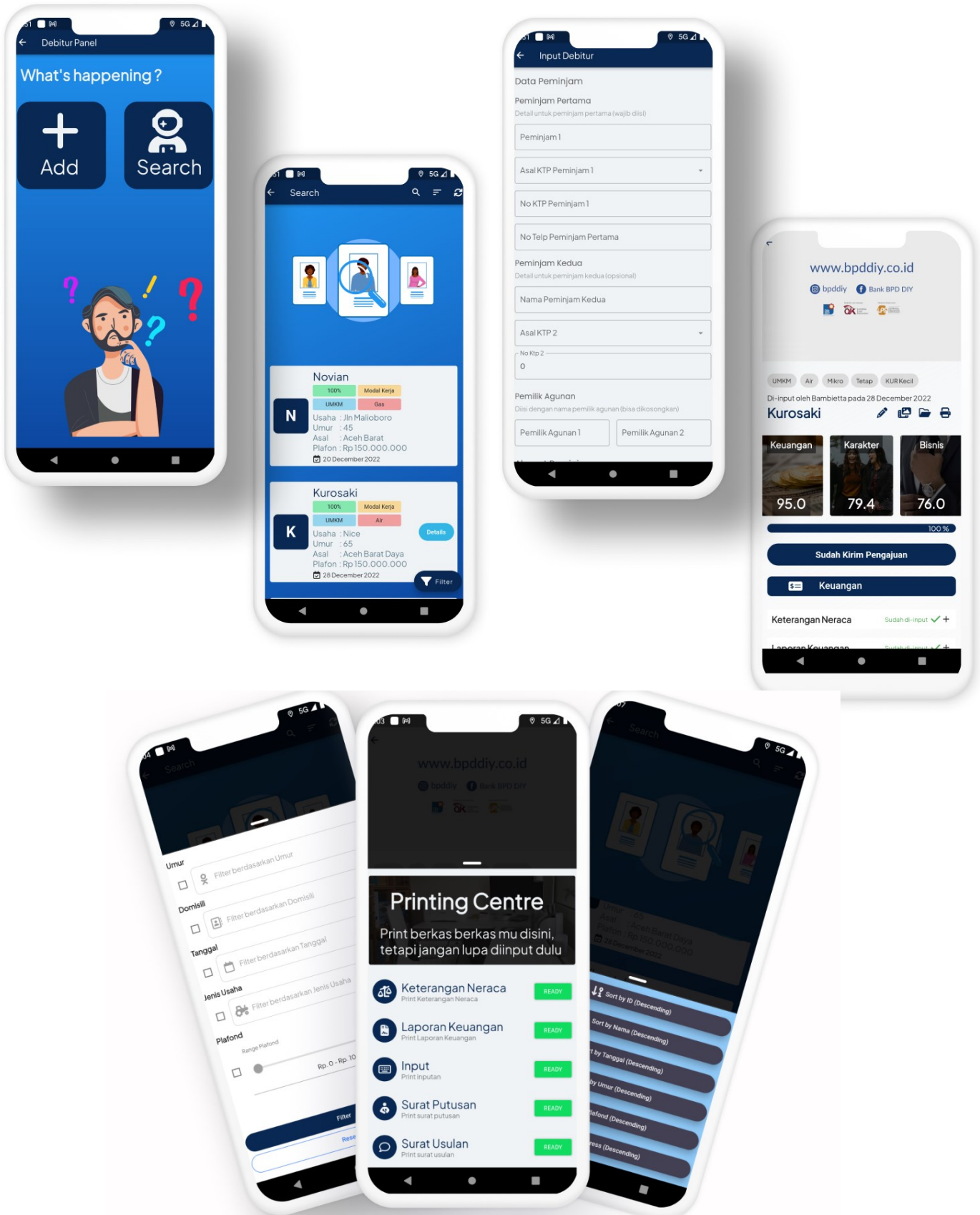
### User Interface Homescreen





## LAMPIRAN #2

### User Interface Mangemen Debitur



**LAMPIRAN #3**  
**User Interface Upload Photo / Files**

