

## **Bab VI**

### **KESIMPULAN**

Dalam bab penutup akan membahas penjelasan mengenai kesimpulan dan saran dari aplikasi yang dibangun, daftar pustaka yang digunakan serta lampiran lampiran.

#### **6.1 Kesimpulan**

Berdasarkan analisis, desain, implementasi, dan pengujian aplikasi maka dapat ditarik kesimpulan bahwa:

1. Aplikasi dapat mempertemukan secara tidak langsung antara petani dan investor dalam suatu aplikasi sekaligus.
2. Bagi petani aplikasi dapat memberikan laporan proses proyek dan bagi investor dapat memantau investasinya.

#### **6.2 Saran**

Berberapa saran dan masukan penulis guna pengembangan Aplikasi Investasi Pertanian Mobile dimasa yang akan datang, meliputi:

1. Ditambahkan fungsionalitas notifikasi untuk mempermudah pengelolaan maupun transaksi.
2. Ditambahkan fungsionalitas pembayaran online untuk memudahkan pengelolaan transaksi.
3. Adanya pengembangan untuk sistem operasi IOS.

## DAFTAR PUSTAKA

- BPS. (2015). *Analisis PDB Sektor Pertanian Tahun 2015*. Kementran Pertanian.
- Developer Android. (2017, Desember). *Android, the world's most popular mobile platform*. Retrieved 12 3, 2017, from Android.com: <https://developer.android.com/about/android.html>
- Developer Android. (2017, Desember). *Dasar-Dasar Aplikasi*. Retrieved 12 3, 2017, from Android.com: <https://developer.android.com/guide/components/fundamentals.html>
- DiMarzio, J. F. (2017). *Beginning Android Programming with Android Studio* (4 ed.). Indianapolis, Indiana: John Wiley & Sons, Inc.
- Drake, J. J., Lanier, Z., Mulliner, C., Olivia, P., Ridley, S. A., & Wicherski, G. (2014). *Android™ Hacker's Handbook*. John Wiley & Sons, Inc.
- Easwaran, R. S., & Ramasundaram, P. (2008). Whether Commodity Futures Market in Agriculture is Efficient in Price Discovery ? — An Econometric Analysis. *Agricultural Economics Research Review*, 21, 337-344.
- eTransformAfrica. (2012). *The Transformational Use of Information and Communication Technologies in Africa*. Retrieved from African Development Bank Group: [https://www.afdb.org/fileadmin/uploads/afdb/Documents/Publications/The\\_Transformational\\_Use\\_of\\_Information\\_and\\_Communication\\_Technologies\\_in\\_Africa.pdf](https://www.afdb.org/fileadmin/uploads/afdb/Documents/Publications/The_Transformational_Use_of_Information_and_Communication_Technologies_in_Africa.pdf)
- Fahmi, I. (2015). *Manajemen Investasi Edisi 2*. Jakarta: Salemba Empat.
- Goenadi, D. H. (2016). Perkembangan, Prospek, dan Kebijakan Penanaman Modal di Sektor Pertanian. *Prosiding Seminar Nasional "Era Baru Pembangunan Pertanian: Strategi Mengatasi Masalah Pangan, Bioenergi dan Perubahan Iklim" 2011* (pp. 28-34). Pusat Sosial Ekonomi dan Kebijakan Pertanian Kementerian Pertanian.
- Hasanuzzaman, M. (2015). *Agriculture and Agronomy*. Retrieved from [www.hasanuzzaman.webs.com](http://www.hasanuzzaman.webs.com): [https://hasanuzzaman.weebly.com/uploads/9/3/4/0/934025/introduction\\_to\\_agriculture\\_and\\_agronomy.pdf](https://hasanuzzaman.weebly.com/uploads/9/3/4/0/934025/introduction_to_agriculture_and_agronomy.pdf)
- Jones, C. P. (2014). *Investments Principles and Concepts* (12 ed.). Wiley.
- Kementrian Pertanian. (2015). *Rencana Strategis Kementerian Pertanian Tahun 2015 - 2019*. Kementerian Pertanian Republik Indonesia 2015.

- Mintert, J., & Welch, M. (n.d.). *Introduction To Futures Markets*. Texax A&M AgriLife Extension.
- Olson, K. D. (2010). *Economics of Farm Management in a Global Setting*. John Wiley & Sons, Inc.
- Poitras, G. (2016). *Futures Markets and Forward Markets*.
- Rizky, R. L., Agustin, G., & Mukhlis, I. (2016). Pengaruh Penanaman Modal Asing, Penanaman Modal Dalam Negeri Dan Belanja Modal Terhadap Pertumbuhan Ekonomi Provinsi Di Indonesia. *Jurnal Ekonomi & Studi Pembangunan*, 8, 9-16.
- Rosalina, Handojo, A., & Wibowo, A. (2015). Aplikasi Crowdfunding Sebagai Perantara Penggalangan. *Jurnal Infra*, 3(2), pp-303.
- Rubin, J. Z., & Brown, B. R. (2013). *The Social Psychology of Bargaining and Negotiation*. Elsevier.
- TaniFund. (2017, Desember). *Menciptakan kesejahteraan bagi petani Indonesia*. Retrieved 12 3, 2017, from TaniFund.com: <https://tanifund.com/about-us>
- TaniFund. (2017, Desember). *TaniFund FAQ*. Retrieved 12 3, 2017, from TaniFund.com: <https://tanifund.com/faq>
- Wahyuningsih, T. (2011). SISTEM BAGI HASIL MARO SEBAGAI UPAYA MEWUJUDKAN SOLIDARITAS MASYARAKAT. *Jurnal Komunitas*, 3(2), 197-204.
- Wibowo, H., Prasetyo, Y. A., & Al-Anshary, F. M. (2015). Pengembangan Modul UKM Web Portal Crowdfunding dengan Metode Prototype dan Framework Codeigniter. *e-Proceeding of Engineering*, 2(2), 5125-5135.
- Widodo, W. D. (2014). Dasar-Dasar Budidaya Tanaman. In *Sistem Pertanian* (pp. 1-91). Jakarta: Universitas Terbuka.
- Winanti, M. B., & Prayogo, D. (2013). PERANCANGAN SISTEM INFORMASI PENANAMAN MODAL DI BADAN KOORDINASI PROMOSI DAN PENANAMAN MODAL DAERAH (BKPPMD) PROVINSI JAWA BARAT. *Jurnal Manajemen Informatika (JAMIKA)*, 03, 1-12.

**Pengembangan Aplikasi Mobile  
Investasi Pertanian  
MIP**

**Spesifikasi Kebutuhan Perangkat Lunak**

**Esthy Yustiana Wahyu / 140707696**

**Dibuat untuk:  
Universitas Atma Jaya Yogyakarta  
2018**

## Daftar Revisi

Tanggal	Deskripsi Revisi	Penulis	Keterangan

## Persetujuan Dokumen

Dokumen Spesifikasi Kebutuhan Perangkat Lunak ini telah diterima dan disetujui oleh:

Nama	Tanggal	Tandatangan

# Daftar Isi

DAFTAR REVISI .....	II
TUJUAN DOKUMEN .....	II
PENDAHULUAN .....	1
1 TUJUAN PENULISAN DOKUMEN .....	1
2 RUANG LINGKUP .....	1
3 DEFINISI, AKRONIM, DAN SINGKATAN .....	2
4 REFERENSI .....	2
5 IKHTISAR DOKUMEN .....	2
DESKRIPSI UMUM KEBUTUHAN .....	3
1 PERSPEKTIF PRODUK .....	3
2 FUNGSI PRODUK .....	4
3 KARAKTERISTIK PENGGUNA .....	4
4 KEKANGAN .....	4
5 ASUMSI DAN KEBERGANTUNGAN .....	5
KEBUTUHAN RINCI .....	5
1 KEBUTUHAN ANTARMUKA .....	5
3.1.1 Antarmuka Pengguna .....	5
3.1.2 Antarmuka Perangkat Keras .....	6
3.1.3 Antarmuka Perangkat Lunak .....	6
2 KEBUTUHAN FUNGSIONAL .....	7
3.2.1 Fungsi Login Aplikasi .....	7
3.2.2 Fungsi Registrasi Aplikasi .....	7
3.2.3 Fungsi Pengelolaan Proyek .....	7
3.2.4 Fungsi Pengelolaan Proses Proyek .....	8
3.2.5 Fungsi Tampil Daftar Proyek Pertanian .....	8
3.2.6 Fungsi Pendanaan Proyek Pertanian .....	8
3.2.7 Fungsi Menawar Hasil Bagi .....	8
3.2.8 Fungsi Upload Bukti Pembayaran Pendanaan Proyek Pertanian .....	9
3.2.9 Fungsi Verifikasi Pendanaan Proyek Pertanian .....	9
3.2.10 Fungsi Tampil Laporan Progress Proyek Pertanian .....	9
3.2.11 Fungsi Pengelolaan Portofolio Petani .....	9
3 DIAGRAM USE CASE .....	10
4 KEBUTUHAN DATA .....	21

## **. Pendahuluan**

### **.1 Tujuan Penulisan Dokumen**

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) ini merupakan dokumen spesifikasi kebutuhan perangkat lunak aplikasi mobile investasi pertanian untuk mendefinisikan kebutuhan perangkat lunak yang meliputi antarmuka eksternal (antarmuka antara sistem dengan sistem lain perangkat lunak dan perangkat keras, dan pengguna) dan atribut (feature-feature tambahan yang dimiliki sistem), serta mendefinisikan fungsi perangkat lunak. SKPL-MIP ini juga mendefinisikan batasan perancangan perangkat lunak.

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) ini ditulis untuk pengembang selanjutnya, yang akan melanjutkan pengembangan aplikasi investasi pertanian ini.

### **.2 Ruang Lingkup**

Ruang lingkup perangkat lunak ini dikembangkan sebagai berikut:

Perangkat lunak investasi pertanian ini dibangun dengan menggunakan *tools* Android Studio dengan menggunakan database MySQL. Perangkat lunak ini akan berjalan pada lingkungan mobile dengan sistem operasi Android. Perangkat lunak akan berkomunikasi dengan database menggunakan API.

Perangkat lunak investasi pertanian ini dibangun dengan fungsi yaitu pengelolaan proyek pertanian, baik menambah data baru, mengubah data, dan menghapusnya. Fungsi lainnya adalah pengelolaan proses pertanian. Fungsi utama dari aplikasi ini adalah fungsi tawar-menawar dan fungsi pendanaan pada suatu proyek. Perangkat lunak ini tidak dapat menangani transaksi pembayaran secara online, juga tidak menangani proses akhir pengembalian keuntungan dari petani ke investornya.

Perangkat Lunak ini dikembangkan dengan sasaran guna membantu petani Indonesia untuk mendapatkan modal lebih mudah dengan cara membuka peluang investasi untuk produk pertanian maupun peternakan mereka.

### 1.3 Definisi, Akronim, dan Singkatan

Definisi	
Investor	Orang perorangan atau lembaga baik domestik atau non domestik yang melakukan suatu investasi (bentuk penanaman modal sesuai dengan jenis investasi yang dipilihnya) baik dalam jangka pendek atau jangka panjang.
Petani	Seseorang yang bergerak di bidang pertanian, dengan cara melakukan pengelolaan tanah dengan tujuan untuk menumbuhkan dan memelihara tanaman, dengan harapan untuk memperoleh hasil.
Tawar-Menawar	Proses dimana dua atau lebih pihak berusaha untuk menyelesaikan apa yang masing-masing akan beri dan akan menerima, atau yang masing-masing lakukan dan terima dalam suatu transaksi diantara mereka

Akronim dan Singkatan	
SKPL	Spesifikasi Kebutuhan Perangkat Lunak
MIP	Mobile Investasi Pertanian
SKPL-MIP-XXX-YY	Kode yang merepresentasikan kebutuhan pada aplikasi dimana XXX merupakan nomor fungsi produk dan YY merupakan subfungsi produk.

### 1.4 Referensi

Referensi yang digunakan pada perangkat lunak yang dikembangkan adalah berikut:

Wahyu, Esthy Yustiana (2016). *SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK SNBC*.

Yogyakarta: Universitas Atma Jaya Yogyakarta.

Lestari, F. Y. (2017). *SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK PTT*. Yogyakarta:

Universitas Atma Jaya Yogyakarta.

### 1.5 Ikhtisar Dokumen

Secara umum dokumen SKPL ini terbagi atas 5 bagian utama.

Bagian utama berisi penjelasan mengenai dokumen SKPL tersebut yang mencakup tujuan pembuatan SKPL, ruang lingkup dalam pengembangan perangkat lunak, definisi, referensi dan ikhtisar dokumen tentang dokumen SKPL ini.



Bagian kedua berisi penjelasan umum tentang perangkat lunak aplikasi mobile yang akan dikembangkan, mencakup perspektif produk yang akan dikembangkan, fungsi produk perangkat lunak, karakteristik pengguna, batasan dalam penggunaan perangkat lunak dan asumsi yang dipakai dalam pengembangan perangkat lunak tersebut.

Bagian ketiga berisi penjelasan secara lebih rinci tentang kebutuhan perangkat lunak yang akan dikembangkan dan spesifikasi kebutuhan fungsionalitas perangkat lunak yang akan dikembangkan serta gambar ERD (Entity Relationship Diagram) dari skema yang dibutuhkan untuk membuat database perangkat lunak.

## **2. Deskripsi Umum Kebutuhan**

### **2.1 Perspektif Produk**

Aplikasi Investasi Pertanian ini merupakan perangkat lunak yang dikembangkan untuk membantu pengelolaan sistem informasi pada mobile. Sistem ini dibuat untuk membantu petani mendapatkan investor melalui aplikasi ini. Sistem ini dapat dipergunakan oleh Petani sendiri maupun dipergunakan oleh Investor dan sistem ini sendiri akan dibangun pada aplikasi mobile dengan platform android.

Pada aplikasi mobile digunakan oleh pihak pencari modal seperti petani dan pihak investor. Aplikasi ini menangani antara lain: Pengelolaan layanan investasi berbasis teknologi informasi yang meliputi layanan pendaftaran proyek pertanian, layanan pendaftaran investor, pemilihan proyek pertanian, dan pemberian laporan proyek pertanian. Pengelolaan data pendukung proses bisnis yang meliputi pengelolaan data proyek pertanian.

Perangkat lunak mobile ini berjalan pada platform android. Bahasa pemrograman yang digunakan adalah bahasa java, serta XML dalam pembuatan layout. Tools yang digunakan untuk pengembangan menggunakan Android Studio. Database yang akan digunakan adalah MySQL. Untuk mengakses ke database menggunakan API.

## 2.2 Fungsi Produk

### 1. Proses Pengolahan Data

Deskripsi	:	Fungsi ini memungkinkan petani untuk memasukkan data proyek. Data proyek yang dimasukkan berupa nama produk, deskripsi produk, jenis produk, judul proyek, luas lahan, target proyek, tanggal dimulai dan selesai proyek, dan jumlah dana yang dibutuhkan.
Rasional	:	Fungsi ini digunakan untuk membantu petani dalam proses penginputan data proyek agar lebih menarik untuk investor.

### 2. Proses Tawar dan Transaksi Pendanaan

Deskripsi	:	Fungsi ini memungkinkan investor untuk melakukan tawaran untuk persentase hasil bagi dengan petani. Dan juga memungkinkan investor untuk melakukan transaksi investasi atau pendanaan pada proyek yang diinginkan. Transaksi yang dimaksud berupa; penawaran dan pendanaan.
Rasional	:	Fungsi ini diperlukan agar dapat membantu para investor dalam menawar hasil bagi dan melakukan transaksi investasi

### 3. Proses Pelaporan

Deskripsi	:	Fungsi ini memungkinkan investor dan petani untuk mendapatkan laporan berupa list view yang mencakup; list proyek, list tawaran, dan list proses proyek
Rasional	:	Fungsi ini diperlukan untuk melihat semua riwayat transaksi, baik transaksi pendanaan maupun tawaran.

## 2.3 Karakteristik Pengguna

Karakteristik dari pengguna perangkat lunak ini adalah sebagai berikut:

- Memahami penggunaan Internet dan Android versi 4.2 keatas.
- Memahami penggunaan aplikasi mobile Investasi Pertanian.

## 2.4 Kekangan

Batasan dalam penggunaan perangkat lunak ini adalah:

- Tidak adanya koneksi internet

Jika tidak ada koneksi internet yang tesambung, aplikasi tidak dapat mengakses API untuk mendapatkan data yang dibutuhkan oleh aplikasi.

## 2.5 Asumsi dan Kebergantungan

Perangkat lunak ini dapat dijalankan dengan baik perangkat mobile Android dengan minimal sistem operasi Android versi 4.2 (Jelly Bean).

## 3. Kebutuhan Rinci

### 3.1 Kebutuhan Antarmuka

#### 3.1.1 Antarmuka Pengguna

Perangkat Lunak ini memiliki form antar muka berikut :

No	Nama Form	:	Deskripsi Fungsi Form
1	Form Login Petani	:	Digunakan oleh petani untuk masuk ke aplikasi.
2	Form Login Member	:	Digunakan oleh member untuk masuk ke aplikasi
3	Form Register Petani	:	Digunakan oleh petani baru untuk mendaftar ke aplikasi.
4	Form Register Member	:	Digunakan oleh member baru untuk mendaftar ke aplikasi.
5	Form Beranda Petani dan Member	:	Menampilkan beranda petani dan member.
6	Form Katalog Ku (Petani)	:	Menampilkan proyek – proyek yang telah dibuat oleh petani yang sedang dalam aplikasi.
7	Form Detail Katalog (Petani)	:	Menampilkan detail proyek yang dipilih dari form katalogKu. Berisi pilihan untuk ubah dan hapus jika proyek tersebut belum didanai.
8	Form Input Proyek	:	Menginputkan proyek baru.
9	Form Edit Proyek	:	Mengedit proyek lama.
10	Form Katalog Ku (Member)	:	Menampilkan proyek – proyek yang terdapat pada database beserta statusnya.
11	Form Detail Katalog (Member)	:	Menampilkan detail proyek yang dipilih. Terdapat button untuk mendanai proyek tersebut dan button untuk menawarkan hasil bagi pertanian.
12	Form Transaksi Pembayaran Proyek	:	Menginputkan transaksi yang telah dilakukan oleh member.
13	Form List Tawaran (Petani)	:	Menampilkan tawaran dan statusnya berdasarkan id admin.
14	Form List Tawaran (Member)	:	Menampilkan tawaran dan statusnya berdasarkan id member.
15	Form Tawaran (Petani dan Member)	:	Menampilkan persentase hasil petani dan hasil member. Berisi button setuju maupun button tidak setuju.
16	Form Tawaran yg tidak disetujui (Petani dan Member)	:	Menginputkan tawaran baru.

17	Form List Proyek (Petani)	:	Menampilkan proyek – proyek petani yang telah didanai maupun sedang menunggu konfirmasi petani.
18	Form List Proyek (Member)	:	Menampilkan proyek – proyek yang telah didanai member maupun sedang menunggu konfirmasi member.
19	Form Konfirmasi Proyek Petani	:	Mengkonfirmasi proyek yang dananya telah diterima oleh petani.
20	Form Detail Proses (Petani)	:	Menampilkan detail proses dari proyek yang telah dipilih dan didanai dari form prosesKu.
21	Form Input Proses Proyek (Petani)	:	Menginputkan proses dari proyek yang telah dipilih pada list proyek sebelumnya
22	Form Upload Bukti Pendanaan Proyek	:	Menginputkan bukti pendanaan proyek yang didanai.
23	Form Detail Proses (Member)	:	Menampilkan detail proses dari proyek yang telah dipilih dan didanai dari form prosesKu.
24	Form Profil Petani	:	Menampilkan profil petani yang sedang dalam aplikasi dan menampilkan portofolio milik petani pada bagian bawah profil.
25	Form Edit Profil Petani	:	Mengubah data profil petani yang sedang dalam aplikasi.
26	Form Input Portofolio Petani	:	Menginputkan portofolio petani yang sedang log In aplikasi.
27	Form Profil Member	:	Menampilkan profil member yang sedang dalam aplikasi.
28	Form Edit Profil Member	:	Mengubah data profil member yang sedang dalam aplikasi.

### 3.1.2 Antarmuka Perangkat Keras

Antarmuka perangkat keras yang digunakan untuk menjalankan aplikasi adalah :

- a. Perangkat ponsel pintar dengan sistem operasi Android.

### 3.1.3 Antarmuka Perangkat Lunak

Perangkat lunak yang dibutuhkan untuk menjalankan aplikasi adalah :

- a. Nama : Sistem Operasi Android versi 4.2

Sumber : Google

Sebagai sistem operasi untuk menjalankan aplikasi.

- b. Nama : Android Studio

Sumber : Google

Sebagai alat yang digunakan untuk membangun aplikasi dengan menggunakan bahasa pemrograman Java.

- c. Nama : Apache

Sumber : XAMP

Sebagai alat yang digunakan sebagai web server.

## 3.2 Kebutuhan Fungsional

### 3.2.1. Fungsi Login Aplikasi

ID Requirement	:	Login Aplikasi (SKPL-MIP-001)
Deskripsi	:	Merupakan fungsi yang digunakan untuk proses otorisasi oleh semua pengguna yang telah terdaftar untuk bisa masuk kedalam sistem aplikasi.
Validity Check		<ul style="list-style-type: none"><li>▪ Username dan Password tidak boleh kosong.</li><li>▪ Minimal 6 karakter</li></ul>
Rasional	:	Fungsi ini digunakan untuk mengotorisasi pengguna yang sedang berada dalam sistem aplikasi. Sehingga pengguna dapat menggunakan aplikasi sesuai dengan hak akses masing – masing.
Referensi	:	-

### 3.2.2. Fungsi Registrasi Aplikasi

ID Requirement	:	Registrasi Aplikasi (SKPL-MIP-002)
Deskripsi	:	Merupakan fungsi yang digunakan untuk mendaftarkan pengguna untuk bisa masuk ke sistem.
Validity Check		<ul style="list-style-type: none"><li>▪ Nama lengkap, Email, NIK, Username, Password tidak boleh kosong</li><li>▪ Format email harus benar</li><li>▪ NIK berupa angka 16 digit</li><li>▪ Username dan password minimal 6 karakter</li></ul>
Rasional	:	Fungsi ini disediakan untuk pengguna baru maupun member baru yang ingin dapat menggunakan aplikasi sesuai dengan peran sebagai member atau petani.
Referensi	:	-

### 3.2.3. Fungsi Pengelolaan Proyek

ID Requirement	:	Pengelolaan Proyek (SKPL-MIP-003)
Deskripsi	:	Merupakan fungsi yang digunakan oleh Petani untuk mengelola proyeknya. Meliputi menambah, menampilkan, mengubah, dan menghapus data proyek.
Validity Check		<ul style="list-style-type: none"><li>▪ Semua field harus diisi</li><li>▪ Tanggal mulai proyek tidak boleh lebih dari tanggal selesai proyek</li><li>▪ Luas lahan dalam satuan hektar berupa angka</li><li>▪ Jumlah Ternak dalam satuan ekor berupa angka</li><li>▪ Target hasil dalam satuan kilogram / ekor berupa angka</li><li>▪ Jumlah dana yang dibutuhkan dalam bentuk nominal mata uang rupiah</li><li>▪ Hasil Petani dan hasil member dalam satuan persen berupa angka maksimal 2 digit</li><li>▪ Hasil Petani dan hasil member jika dijumlah harus 100%</li></ul>

Rasional	:	Fungsi ini disediakan untuk petani mengelola proyek yang akan didanai oleh member. Proyek dapat meliputi proyek pertanian maupun proyek peternakan. Proyek yang telah didanai tidak dapat hapus.
Referensi	:	-

#### 3.2.4. Fungsi Pengelolaan Proses Proyek

ID Requirement	:	Pengelolaan Proses Proyek (SKPL-MIP-004)
Deskripsi	:	Merupakan fungsi yang digunakan oleh petani untuk mengelola proses dari proyek yang telah didanai. Meliputi menambah, menampilkan, dan mengubah data proses proyek.
Validity Check		<ul style="list-style-type: none"> <li>▪ Semua field harus diisi</li> <li>▪ Tanggal proses merupakan tanggal sekarang</li> </ul>
Rasional	:	Fungsi ini disediakan untuk petani mengelola proses dari proyek yang telah didanai oleh member. Proses proyek yang telah diinputkan tidak dapat hapus.
Referensi	:	-

#### 3.2.5. Fungsi Tampil Daftar Proyek Pertanian

ID Requirement	:	Tampil Display Daftar Proyek Pertanian (SKPL-MIP-005)
Deskripsi	:	Merupakan fungsi yang digunakan untuk menampilkan daftar proyek kepada member yaitu investor yang tersimpan pada database.
Validity Check		<ul style="list-style-type: none"> <li>▪ Tidak ada</li> </ul>
Rasional	:	Fungsi ini disediakan untuk member melihat-lihat proyek yang ada didatabase. Proyek ditampilkan beserta dengan detail untuk menarik minat member untuk mendanai proyek tersebut.
Referensi	:	-

#### 3.2.6. Fungsi Pendanaan Proyek Pertanian

ID Requirement	:	Pendanaan Proyek Pertanian (SKPL-MIP-006)
Deskripsi	:	Merupakan fungsi yang digunakan oleh Investor dalam hal ini member untuk melakukan pendanaan ke proyek tertentu.
Validity Check		<ul style="list-style-type: none"> <li>▪ Jumlah dana yang dimasukkan tidak boleh kurang dari harga proyek tersebut</li> <li>▪ Inputan jumlah dana berupa angka desimal</li> </ul>
Rasional	:	Fungsi ini disediakan untuk member yang tertarik pada suatu proyek tertentu untuk mendanai proyek tersebut dengan kesepakatan bagi hasil yang telah ditentukan.
Referensi	:	-

#### 3.2.7. Fungsi Menawar Hasil Bagi

ID Requirement	:	Menawar Hasil Bagi (SKPL-MIP-007)
Deskripsi	:	Merupakan fungsi yang digunakan oleh Investor dalam hal ini member untuk menawar persentase bagi hasil yang telah ditawarkan oleh petani.
Validity Check		<ul style="list-style-type: none"> <li>▪ Jumlah persentase hasil member dan hasil petani harus 100%</li> </ul>

	▪ Hasil member dan hasil petani berupa angka desimal maksimal 2 digit
Rasional	: Fungsi ini disediakan untuk member yang tertarik pada suatu proyek tertentu dan ingin melakukan transaksi investasi untuk memilih terlebih dahulu persentasi bagi hasil dengan petani / petani.
Referensi	: -

### 3.2.8. Fungsi Upload Bukti Pembayaran Pendanaan Proyek Pertanian

ID Requirement	: Upload Bukti Pembayaran Pendanaan Proyek Pertanian (SKPL-MIP-008)
Deskripsi	: Merupakan fungsi yang digunakan oleh Investor untuk mengupload bukti transfer proyek pertanian.
Validity Check	▪ Gambar tidak boleh kosong
Rasional	: Fungsi ini disediakan untuk member yang telah melakukan transaksi pembayaran melalui transfer ke bank terkait untuk mengkonfirmasi pembayaran dengan cara mengupload bukti transfer.
Referensi	: -

### 3.2.9. Fungsi Verifikasi Pendanaan Proyek Pertanian

ID Requirement	: Verifikasi Pendanaan Proyek Pertanian (SKPL-MIP-009)
Deskripsi	: Merupakan fungsi yang digunakan oleh Petani untuk memverifikasi dana investor yang telah masuk.
Validity Check	▪ Bukti upload pembayaran harus sudah ditampilkan
Rasional	: Fungsi ini disediakan untuk petani mengkonfirmasi bahwa dana telah diterima oleh petani dan memulai proyek yang telah didanai.
Referensi	: -

### 3.2.10. Fungsi Tampil Laporan Progress Proyek Pertanian

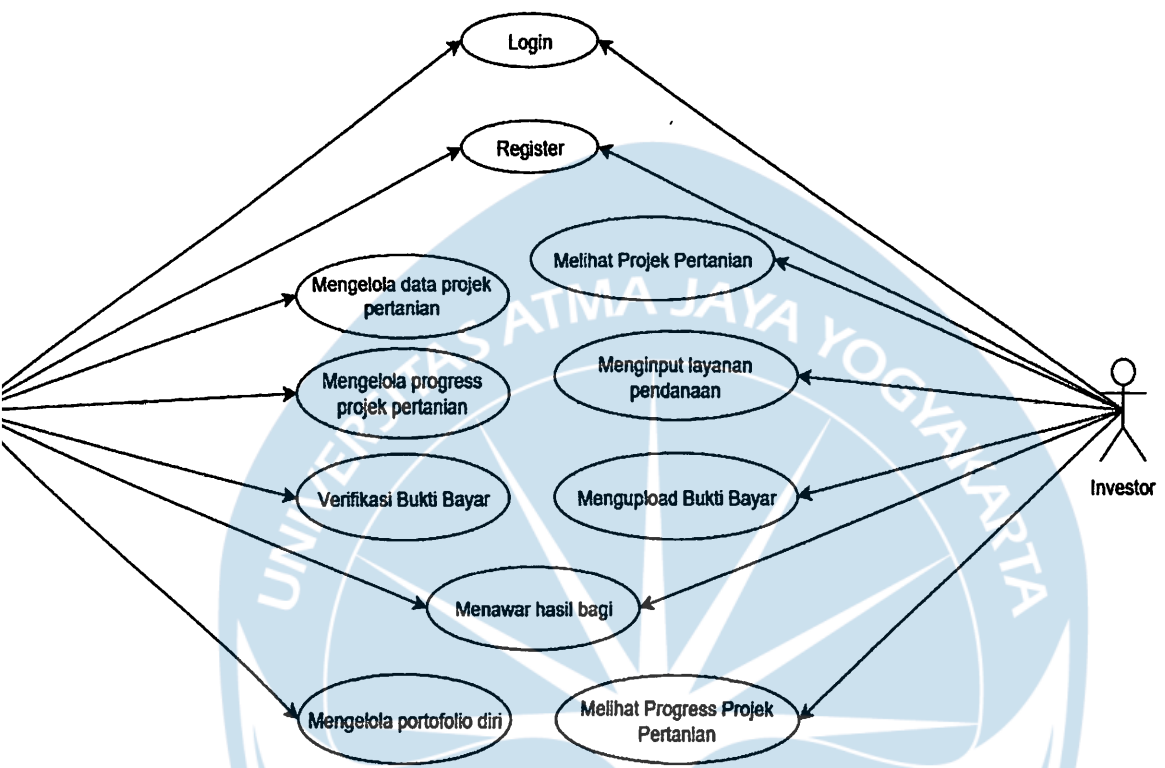
ID Requirement	: Tampil Laporan Progress Proyek Pertanian (SKPL-MIP-010)
Deskripsi	: Merupakan fungsi yang digunakan oleh Investor untuk melihat laporan progress proyek pertanian yang telah didanai.
Validity Check	▪ Tidak Ada
Rasional	: Fungsi ini disediakan untuk member memantau perkembangan proyek yang telah didanainya.
Referensi	: -

### 3.2.11. Fungsi Pengelolaan Portofolio Petani

ID Requirement	: Pengelolaan Portofolio Petani (SKPL-MIP-011)
Deskripsi	: Merupakan fungsi yang digunakan oleh Petani untuk mengelola portofolio diri mereka.
Validity Check	<ul style="list-style-type: none"> <li>▪ Semua field harus diisi</li> <li>▪ Luas lahan / jumlah ternak dalam satuan hektar / ekor</li> <li>▪ Hasil panen / ternak dalam satuan kg / ekor</li> </ul>

isional	:	Fungsi ini disediakan untuk petani mengelola portofolio diri mereka agar investor semakin yakin dengan kinerja petani.
erensi	:	-

### Diagram Use Case



Gambar 1 Usecase Diagram Aplikasi

nama Use Case	:	Login Member Aplikasi sebagai member
deskripsi singkat	:	Usecase ini digunakan oleh aktor untuk masuk ke dalam aplikasi.
aktor	:	Member
prerequisite	:	Aktor sudah terdaftar kedalam database sistem aplikasi.
post kondisi	:	Aktor masuk kedalam sistem aplikasi sebagai Member.
alir Flow	:	<ol style="list-style-type: none"> <li>1. Use case ini dimulai ketika aktor memilih untuk melakukan login sebagai member ke aplikasi.</li> <li>2. Sistem menampilkan form login.</li> <li>3. Aktor memasukkan username dan password kemudian mengklik tombol login.</li> <li>4. Sistem mengecek validasi username dan password aktor pada database sistem.</li> </ol> <p>E-1 Username atau password yang dimasukkan oleh aktor salah</p> <ol style="list-style-type: none"> <li>5. Sistem menampilkan halaman utama, aktor dapat masuk ke aplikasi</li> </ol>
alternative Flow	:	None
error Flow	:	E-1 Username atau password yang dimasukkan oleh aktor salah



	<ul style="list-style-type: none"> <li>a. Sistem memberikan pesan peringatan bahwa username atau password yang dimasukkan oleh aktor salah.</li> <li>b. Kembali ke Basic Flow langkah 3.</li> </ul>
--	---

<b>Nama Use Case</b>	: Login petani Aplikasi
<b>Deskripsi singkat</b>	: Usecase ini digunakan oleh aktor untuk masuk ke dalam aplikasi sebagai petani atau pengelola
<b>Aktor</b>	: Petani
<b>Pre kondisi</b>	: Aktor sudah terdaftar kedalam database sistem aplikasi.
<b>Post Kondisi</b>	: Aktor masuk kedalam sistem aplikasi sebagai Member.
<b>Basic Flow</b>	: <ul style="list-style-type: none"> <li>1. Use case ini dimulai ketika aktor memilih untuk melakukan login sebagai petani ke aplikasi.</li> <li>2. Sistem menampilkan form login.</li> <li>3. Aktor memasukkan username dan password kemudian mengklik tombol login.</li> <li>4. Sistem mengecek validasi username dan password aktor pada database sistem.</li> <li>E-1 Username atau password yang dimasukkan oleh aktor salah</li> <li>5. Sistem menampilkan halaman utama, aktor dapat masuk ke aplikasi</li> </ul>
<b>Alternative Flow</b>	: None
<b>Error Flow</b>	: <ul style="list-style-type: none"> <li>E-1 Username atau password yang dimasukkan oleh aktor salah <ul style="list-style-type: none"> <li>a. Sistem memberikan pesan peringatan bahwa username atau password yang dimasukkan oleh aktor salah.</li> <li>b. Kembali ke Basic Flow langkah 3.</li> </ul> </li> </ul>

<b>Nama Use Case</b>	: Registrasi Member Aplikasi
<b>Deskripsi singkat</b>	: Usecase ini digunakan oleh aktor untuk melakukan registrasi aplikasi sebagai member
<b>Aktor</b>	: Member
<b>Pre kondisi</b>	: Aktor belum terdaftar kedalam database sistem aplikasi.
<b>Post Kondisi</b>	: Aktor terdaftar dalam sistem dan masuk kedalam sistem aplikasi sebagai Member
<b>Basic Flow</b>	: <ul style="list-style-type: none"> <li>1. Use case ini dimulai ketika aktor memilih untuk melakukan registrasi sebagai member ke aplikasi.</li> <li>2. Sistem menampilkan form registrasi member.</li> <li>3. Aktor memasukkan data member kemudian mengklik tombol registrasi.</li> <li>4. Sistem mengecek validasi data member pada database sistem.</li> <li>E-1 Data Member yang dimasukkan oleh aktor salah</li> <li>5. Sistem menyimpan data member ke sistem database</li> </ul>
<b>Alternative Flow</b>	: None
<b>Error Flow</b>	: <ul style="list-style-type: none"> <li>E-1 Data Member yang dimasukkan oleh aktor salah <ul style="list-style-type: none"> <li>a. Sistem memberikan pesan peringatan bahwa data member yang dimasukkan oleh aktor salah.</li> </ul> </li> </ul>

b. Kembali ke Basic Flow langkah 3.

Nama Use Case	:	Registrasi Petani Aplikasi
Deskripsi singkat	:	Usecase ini digunakan oleh aktor untuk melakukan registrasi aplikasi sebagai petani
Aktor	:	Petani
Pre kondisi	:	Aktor belum terdaftar kedalam database sistem aplikasi.
Pos Kondisi	:	Aktor terdaftar dalam sistem dan masuk kedalam sistem aplikasi sebagai petani
Basic Flow	:	<ol style="list-style-type: none"> <li>1. Use case ini dimulai ketika aktor memilih untuk melakukan registrasi sebagai petani ke aplikasi.</li> <li>2. Sistem menampilkan form registrasi petani.</li> <li>3. Aktor memasukkan data petani kemudian mengklik tombol registrasi.</li> <li>4. Sistem mengecek validasi data petani pada database sistem.</li> </ol> <p>E-1 Data petani yang dimasukkan oleh aktor salah</p> <ol style="list-style-type: none"> <li>5. Sistem menyimpan data petani ke sistem database</li> </ol>
Alternative Flow	:	None
Error Flow	:	<p>E-1 Data petani yang dimasukkan oleh aktor salah</p> <ol style="list-style-type: none"> <li>a. Sistem memberikan pesan peringatan bahwa data member yang dimasukkan oleh aktor salah.</li> <li>b. Kembali ke Basic Flow langkah 3.</li> </ol>

Nama Use Case	:	Pengelolaan Data Proyek Pertanian
Deskripsi singkat	:	Usecase ini digunakan oleh aktor untuk melakukan pengelolaan data proyek pertanian. Aktor dapat melakukan pendaftaran proyek pertanian, mengubah data proyek pertanian, menampilkan data proyek pertanian, dan menghapus data proyek pertanian.
Aktor	:	Petani
Pre kondisi	:	<ol style="list-style-type: none"> <li>1. Aktor telah melakukan login sebagai petani.</li> <li>2. Aktor telah memasuki sistem.</li> </ol>
Pos Kondisi	:	Data proyek pertanian di database telah ter update
Basic Flow	:	<ol style="list-style-type: none"> <li>1. Use case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data proyek pertanian.</li> <li>2. Sistem memberikan pilihan untuk melakukan pendaftaran proyek pertanian, ubah proyek pertanian, tampil proyek pertanian dan menghapus proyek pertanian.</li> <li>3. Aktor memilih untuk melakukan pendaftaran proyek pertanian.             <ul style="list-style-type: none"> <li>A-1 Aktor memilih untuk melakukan ubah data proyek pertanian</li> <li>A-2 Aktor memilih untuk melakukan tampil data proyek pertanian</li> <li>A-3 Aktor memilih untuk melakukan hapus data proyek pertanian</li> </ul> </li> <li>4. Sistem menampilkan form produk pertanian yang menjadi proyek pertanian.</li> <li>5. Aktor mengisi form data produk pertanian, seperti; nama produk, jenis produk, dll.</li> </ol>

	<p>6. Aktor memilih button Next untuk melanjutkan penambahan proyek pertanian.</p> <p>7. Sistem mengecek data produk pertanian yang telah dimasukkan oleh aktor. E-1 Data produk pertanian yang dimasukkan oleh aktor salah</p> <p>8. Sistem menampilkan form proyek pertanian</p> <p>9. Aktor mengisi form data proyek pertanian, seperti; nama proyek, tanggal proyek dimulai dan selesai, dll.</p> <p>10. Aktor memilih button Next untuk melanjutkan penambahan proyek pertanian.</p> <p>11. Sistem mengecek data proyek pertanian yang telah dimasukkan oleh aktor. E-2 Data proyek pertanian yang dimasukkan oleh aktor salah</p> <p>12. Sistem menampilkan form dana proyek pertanian</p> <p>13. Aktor mengisi form dana data proyek pertanian, seperti; jumlah dana yang dibutuhkan, hasil bagi keuntungan, dll.</p> <p>14. Aktor memilih button Add untuk melanjutkan penambahan proyek pertanian.</p> <p>15. Sistem mengecek data dana proyek pertanian yang telah dimasukkan oleh aktor. E-3 Data dana proyek pertanian yang dimasukkan oleh aktor salah</p> <p>16. Aktor meminta sistem untuk menyimpan semua data yang telah dimasukkan.</p> <p>17. Sistem mengecek semua data yang telah dimasukkan oleh aktor E-4 Data yang dimasukkan oleh aktor salah</p> <p>18. Sistem menyimpan semua data ke database dan menampilkan pesan bahwa data proyek pertanian berhasil disimpan</p> <p>19. Use case selesai</p>
Alternative Flow	<p>: A-1 Aktor memilih untuk melakukan ubah data proyek pertanian</p> <ol style="list-style-type: none"> <li>1. Sistem menampilkan list data proyek pertanian</li> <li>2. Aktor memilih data yang akan diubah</li> <li>3. Sistem menampilkan detail proyek pertanian</li> <li>4. Aktor mengubah data proyek pertanian yang sudah dipilih</li> <li>5. Aktor meminta sistem untuk menyimpan data proyek pertanian yang telah diubah</li> <li>6. Sistem melakukan pengecekan terhadap data proyek pertanian yang telah diubah E-5 Data proyek pertanian yang dimasukkan oleh aktor salah</li> <li>7. Sistem menyimpan data proyek pertanian yang telah diubah ke database</li> <li>8. Berlanjut ke Basic flow langkah 19</li> </ol> <p>A-2 Aktor memilih untuk melakukan tampil data proyek pertanian</p> <ol style="list-style-type: none"> <li>1. Sistem menampilkan list data proyek pertanian</li> <li>2. Aktor memilih data yang akan ditampilkan</li> <li>3. Sistem menampilkan detail data proyek pertanian yang dipilih</li> <li>4. Berlanjut ke Basic flow langkah 19</li> </ol> <p>A-3 Aktor memilih untuk melakukan hapus data proyek pertanian</p> <ol style="list-style-type: none"> <li>1. Sistem menampilkan list data proyek pertanian</li> </ol>

	<ol style="list-style-type: none"> <li>2. Aktor memilih data yang akan dihapus</li> <li>3. Sistem menampilkan pesan konfirmasi data yang akan dihapus</li> <li>4. Aktor mengkonfirmasi pesan dengan memilih button Ya</li> <li>5. Sistem menghapus data dari list</li> <li>E-6 Data proyek pertanian masih digunakan oleh sistem</li> <li>6. Sistem menampilkan pesan data berhasil dihapus</li> <li>7. Berlanjut ke Basic flow langkah 15</li> </ol>
Error Flow	<p>:</p> <p>E-1 Data produk pertanian yang dimasukkan oleh aktor salah</p> <ol style="list-style-type: none"> <li>a. Sistem memberikan pesan peringatan bahwa data produk pertanian yang dimasukkan oleh aktor salah.</li> <li>b. Kembali ke Basic Flow langkah 4.</li> </ol> <p>E-2 Data proyek pertanian yang dimasukkan oleh aktor salah</p> <ol style="list-style-type: none"> <li>a. Sistem memberikan pesan peringatan bahwa data proyek pertanian yang dimasukkan oleh aktor salah.</li> <li>b. Kembali ke Basic Flow langkah 8.</li> </ol> <p>E-3 Data dana proyek pertanian yang dimasukkan oleh aktor salah</p> <ol style="list-style-type: none"> <li>a. Sistem memberikan pesan peringatan bahwa data dana proyek pertanian yang dimasukkan oleh aktor salah.</li> <li>b. Kembali ke Basic Flow langkah 12.</li> </ol> <p>E-4 Data yang dimasukkan oleh aktor salah</p> <ol style="list-style-type: none"> <li>a. Sistem memberikan pesan peringatan bahwa data yang dimasukkan oleh aktor salah.</li> <li>b. Kembali ke Basic Flow langkah 12.</li> </ol> <p>E-5 Data proyek pertanian yang dimasukkan oleh aktor salah</p> <ol style="list-style-type: none"> <li>a. Sistem memberikan pesan peringatan bahwa data proyek pertanian yang dimasukkan oleh aktor salah.</li> <li>b. Kembali ke Basic Flow langkah 2.</li> </ol> <p>E-6 Data proyek pertanian masih digunakan oleh sistem</p> <ol style="list-style-type: none"> <li>a. Sistem memberikan pesan peringatan bahwa data dana proyek pertanian tidak dapat dihapus.</li> <li>b. Kembali ke Basic Flow langkah 2.</li> </ol>

Nama Use Case	: Pengelolaan Proses Proyek
Deskripsi singkat	: Usecase ini digunakan oleh aktor untuk melakukan pengelolaan data proses proyek pertanian. Aktor dapat melakukan input proses proyek pertanian, mengubah data proses proyek pertanian, dan menampilkan data proses proyek pertanian.
Aktor	: Petani
Pre kondisi	: <ol style="list-style-type: none"> <li>1. Aktor telah melakukan login sebagai petani.</li> <li>2. Aktor telah memasuki sistem.</li> </ol>
Pos Kondisi	: Data progres proyek pertanian di database telah ter update.
Basic Flow	: <ol style="list-style-type: none"> <li>1. Use case ini dimulai ketika aktor memilih untuk melakukan pengelolaan data progres proyek pertanian.</li> </ol>

	<p>2. Sistem memberikan pilihan untuk melakukan input progres proyek pertanian, ubah progres proyek pertanian, dan tampil progres proyek pertanian.</p> <p>3. Aktor memilih untuk melakukan input progres proyek pertanian</p> <p style="padding-left: 20px;">A-1 Aktor memilih untuk melakukan ubah data progres proyek pertanian</p> <p style="padding-left: 20px;">A-2 Aktor memilih untuk melakukan tampil data progres proyek pertanian</p> <p>4. Sistem menampilkan form input progres proyek pertanian</p> <p>5. Aktor mengisi form data progres proyek pertanian, seperti; tahapan progres, deskripsi progres, tanggal progres, dan lokasi progres, dll.</p> <p>6. Aktor meminta sistem untuk menyimpan data progres proyek pertanian yang telah dimasukkan</p> <p>7. Sistem mengecek data preogres proyek pertanian yang telah dimasukkan oleh aktor</p> <p style="padding-left: 20px;">E-1 Data progres proyek pertanian yang dimasukkan oleh aktor salah</p> <p>8. Sistem menyimpan data progres proyek pertanian ke database dan menampilkan pesan bahwa data progres proyek pertanian berhasil disimpan</p> <p>9. Use case selesai</p>
<b>Alternative Flow</b>	<p>A-1 Aktor memilih untuk melakukan ubah data progres proyek pertanian</p> <ol style="list-style-type: none"> <li>1. Sistem menampilkan list data progres proyek pertanian</li> <li>2. Aktor memilih data yang akan diubah</li> <li>3. Sistem menampilkan detail progres proyek pertanian</li> <li>4. Aktor mengubah data progres proyek pertanian yang sudah dipilih</li> <li>5. Aktor meminta sistem untuk menyimpan data progres proyek pertanian yang telah diubah</li> <li>6. Sistem melakukan pengecekan terhadap data progres proyek pertanian yang telah diubah</li> </ol> <p style="padding-left: 20px;">E-2 Data progres proyek pertanian yang dimasukkan oleh aktor salah</p> <ol style="list-style-type: none"> <li>7. Sistem menyimpan data progres proyek pertanian yang telah diubah ke database</li> <li>8. Berlanjut ke Basic flow langkah 9</li> </ol> <p>A-2 Aktor memilih untuk melakukan tampil data progres proyek pertanian</p> <ol style="list-style-type: none"> <li>1. Sistem menampilkan list data progres proyek pertanian</li> <li>2. Aktor memilih data yang akan ditampilkan</li> <li>3. Sistem menampilkan detail data progres proyek pertanian yang dipilih</li> <li>4. Berlanjut ke Basic flow langkah 9</li> </ol>
<b>Error Flow</b>	<p>E-1 Data progres proyek pertanian yang dimasukkan oleh aktor salah</p> <ol style="list-style-type: none"> <li>a. Sistem memberikan pesan peringatan bahwa data progres proyek pertanian yang dimasukkan oleh aktor salah.</li> <li>b. Kembali ke Basic Flow langkah 4.</li> </ol> <p>E-2 Data progres proyek pertanian yang dimasukkan oleh aktor salah</p> <ol style="list-style-type: none"> <li>a. Sistem memberikan pesan peringatan bahwa data progres proyek pertanian yang dimasukkan oleh aktor salah.</li> <li>b. Kembali ke Basic Flow langkah 3.</li> </ol>

<b>nama Use Case</b>	: Verifikasi Bukti Bayar
<b>deskripsi singkat</b>	: Usecase ini digunakan oleh aktor untuk melakukan verifikasi transaksi pendanaan proyek pertanian.
<b>aktor</b>	: Petani
<b>kondisi</b>	: 1. Aktor telah melakukan login sebagai petani. 2. Aktor telah memasuki sistem.
<b>Kondisi</b>	: Data telah disimpan ke database.
<b>Alir Flow</b>	: 1. Use case ini dimulai ketika aktor memilih tab proses ku 2. Sistem menampilkan list proyek pertanian yang telah diproses beserta status prosesnya. 3. Aktor memilih salah satu proyek pertanian dengan status proses menunggu konfirmasi. 4. Sistem menampilkan proyek pertanian yang menunggu dikonfirmasi. 5. Aktor memilih button Konfirmasi. 6. Sistem menampilkan list proyek pertanian dengan status sudah diproses. 7. Use case selesai.
<b>Alternative Flow</b>	: None
<b>error Flow</b>	: None

<b>nama Use Case</b>	: Menawar Hasil Bagi (Petani)
<b>deskripsi singkat</b>	: Usecase ini digunakan oleh aktor untuk menawar maupun mengkonfirmasi tawaran yang masuk
<b>aktor</b>	: Petani
<b>kondisi</b>	: 1. Aktor telah melakukan login sebagai petani. 2. Aktor telah memasuki sistem.
<b>Kondisi</b>	: Data telah disimpan ke database.
<b>Alir Flow</b>	: 1. Use case ini dimulai ketika aktor memilih tab tawaran ku 2. Sistem menampilkan list tawaran proyek pertanian yang telah diproses beserta status prosesnya. 3. Aktor memilih salah satu item proyek pertanian dengan status proses menunggu. 4. Sistem menampilkan persentase proyek pertanian tersebut. 5. Aktor memilih button Setuju. A-1 Aktor memilih button Tidak Setuju 6. Sistem menampilkan pesan tawaran disetujui. 7. Use case selesai.
<b>Alternative Flow</b>	: A-1 Aktor memilih button Tidak Setuju 1. Sistem menampilkan form baru berisi persentase petani dan member 2. Aktor mengisi form dengan persentase hasil bagi yang baru 3. Sistem mengecek persentase hasil bagi valid atau tidak 4. Aktor memilih button konfirmasi tawaran

Nama Use Case	: Verifikasi Bukti Bayar
Deskripsi singkat	: Usecase ini digunakan oleh aktor untuk melakukan verifikasi transaksi pendanaan proyek pertanian.
Aktor	: Petani
Pre kondisi	: 1. Aktor telah melakukan login sebagai petani. 2. Aktor telah memasuki sistem.
Pos Kondisi	: Data telah disimpan ke database.
Basic Flow	: 1. Use case ini dimulai ketika aktor memilih tab proses ku 2. Sistem menampilkan list proyek pertanian yang telah diproses beserta status prosesnya. 3. Aktor memilih salah satu proyek pertanian dengan status proses menunggu konfirmasi. 4. Sistem menampilkan proyek pertanian yang menunggu dikonfirmasi. 5. Aktor memilih button Konfirmasi. 6. Sistem menampilkan list proyek pertanian dengan status sudah diproses. 7. Use case selesai.
Alternative Flow	: None
Error Flow	: None

Nama Use Case	: Menawar Hasil Bagi (Petani)
Deskripsi singkat	: Usecase ini digunakan oleh aktor untuk menawar maupun mengkonfirmasi tawaran yang masuk
Aktor	: Petani
Pre kondisi	: 1. Aktor telah melakukan login sebagai petani. 2. Aktor telah memasuki sistem.
Pos Kondisi	: Data telah disimpan ke database.
Basic Flow	: 1. Use case ini dimulai ketika aktor memilih tab tawaran ku 2. Sistem menampilkan list tawaran proyek pertanian yang telah diproses beserta status prosesnya. 3. Aktor memilih salah satu item proyek pertanian dengan status proses menunggu. 4. Sistem menampilkan persentase proyek pertanian tersebut. 5. Aktor memilih button Setuju. A-1 Aktor memilih button Tidak Setuju 6. Sistem menampilkan pesan tawaran disetujui. 7. Use case selesai.
Alternative Flow	: A-1 Aktor memilih button Tidak Setuju 1. Sistem menampilkan form baru berisi persentase petani dan member 2. Aktor mengisi form dengan persentase hasil bagi yang baru 3. Sistem mengecek persentase hasil bagi valid atau tidak 4. Aktor memilih button konfirmasi tawaran

<b>Nama Use Case</b>	: Verifikasi Bukti Bayar
<b>Deskripsi singkat</b>	: Usecase ini digunakan oleh aktor untuk melakukan verifikasi transaksi pendanaan proyek pertanian.
<b>Aktor</b>	: Petani
<b>Pre kondisi</b>	: 1. Aktor telah melakukan login sebagai petani. 2. Aktor telah memasuki sistem.
<b>Pos Kondisi</b>	: Data telah disimpan ke database.
<b>Basic Flow</b>	: 1. Use case ini dimulai ketika aktor memilih tab proses ku 2. Sistem menampilkan list proyek pertanian yang telah diproses beserta status prosesnya. 3. Aktor memilih salah satu proyek pertanian dengan status proses menunggu konfirmasi. 4. Sistem menampilkan proyek pertanian yang menunggu dikonfirmasi. 5. Aktor memilih button Konfirmasi. 6. Sistem menampilkan list proyek pertanian dengan status sudah diproses. 7. Use case selesai.
<b>Alternative Flow</b>	: None
<b>Error Flow</b>	: None

<b>Nama Use Case</b>	: Menawar Hasil Bagi (Petani)
<b>Deskripsi singkat</b>	: Usecase ini digunakan oleh aktor untuk menawar maupun mengkonfirmasi tawaran yang masuk
<b>Aktor</b>	: Petani
<b>Pre kondisi</b>	: 1. Aktor telah melakukan login sebagai petani. 2. Aktor telah memasuki sistem.
<b>Pos Kondisi</b>	: Data telah disimpan ke database.
<b>Basic Flow</b>	: 1. Use case ini dimulai ketika aktor memilih tab tawaran ku 2. Sistem menampilkan list tawaran proyek pertanian yang telah diproses beserta status prosesnya. 3. Aktor memilih salah satu item proyek pertanian dengan status proses menunggu. 4. Sistem menampilkan persentase proyek pertanian tersebut. 5. Aktor memilih button Setuju. A-1 Aktor memilih button Tidak Setuju 6. Sistem menampilkan pesan tawaran disetujui. 7. Use case selesai.
<b>Alternative Flow</b>	: A-1 Aktor memilih button Tidak Setuju 1. Sistem menampilkan form baru berisi persentase petani dan member 2. Aktor mengisi form dengan persentase hasil bagi yang baru 3. Sistem mengecek persentase hasil bagi valid atau tidak 4. Aktor memilih button konfirmasi tawaran



	5. Sistem menampilkan list tawaran yang baru 6. Belrlanjut ke basic flow langkah 7
<b>Error Flow</b>	: None

<b>Nama Use Case</b>	: Mengelola Portofolio Diri
<b>Deskripsi singkat</b>	: Usecase ini digunakan oleh aktor untuk mengelola portofolio diri.
<b>Aktor</b>	: Petani
<b>Pre kondisi</b>	: 1. Aktor telah melakukan login sebagai petani. 2. Aktor telah memasuki sistem.
<b>Pos Kondisi</b>	: Data telah disimpan ke database.
<b>Basic Flow</b>	: 1. Use case ini dimulai ketika aktor memilih tab profil ku 2. Sistem menampilkan data diri aktor dan list portofolio dari aktor. 3. Aktor memilih button menambahkan portofolio baru. 4. Sistem menampilkan form portofolio. 5. Aktor mengisi form portofolio seperti; judul portofolio, deskripsi portofolio, luas lahan dan hasil. 6. Sistem mengecek validasi inputan. 7. Aktor memilih button Tambah Portofolio. 8. Sistem menyimpan data ke database dan menampilkan portofolio yang baru ke dalam list.
<b>Alternative Flow</b>	: None
<b>Error Flow</b>	: None

<b>Nama Use Case</b>	: Tampil Display Daftar Proyek Pertanian
<b>Deskripsi singkat</b>	: Usecase ini digunakan oleh aktor untuk melihat semua proyek pertanian yang ada pada database
<b>Aktor</b>	: Member
<b>Pre kondisi</b>	: 1. Aktor telah melakukan login sebagai member. 2. Aktor telah memasuki sistem.
<b>Pos Kondisi</b>	: Data proyek pertanian di database telah ter ditampilkan.
<b>Basic Flow</b>	: 1. Use case ini dimulai ketika aktor memilih tab Catalog. 2. Sistem menampilkan daftar proyek pertanian yang ada pada database. 3. Aktor memilih salah satu proyek pertanian 4. Sistem menampilkan detail proyek pertanian 5. Use case selesai
<b>Alternative Flow</b>	: None
<b>Error Flow</b>	: None

<b>Nama Use Case</b>	: Pendanaan Proyek Pertanian
----------------------	------------------------------

<b>Deskripsi singkat</b>	: Usecase ini digunakan oleh aktor untuk melakukan transaksi pendanaan proyek pertanian.
<b>Aktor</b>	: Member
<b>Prere-kondisi</b>	: 1. Aktor telah melakukan login sebagai member. 2. Aktor telah memasuki sistem.
<b>Post Kondisi</b>	: Data telah disimpan ke database.
<b>Basic Flow</b>	: 1. Use case ini dimulai ketika aktor memilih tab Catalog. 2. Sistem menampilkan daftar proyek pertanian yang ada pada database. 3. Aktor memilih salah satu proyek pertanian. 4. Sistem menampilkan detail proyek pertanian. 5. Aktor memilih button Invest. 6. Sistem menampilkan detail dana proyek pertanian, berupa; jumlah dana yang dibutuhkan, pembagian hasil dengan petani, dll. 7. Aktor memilih option bagi hasil yang diberikan petani. 8. Aktor memasukan nominal dana yang akan diinvestasikan. 9. Sistem mengecek validasi nominal dana. E-1 Nominal dana yang dimasukkan salah. 10. Aktor memilih button Pembayaran 11. Sistem menampilkan cara pembayaran dana proyek pertanian 12. Aktor memilih salah satu metode pembayaran. 13. Sistem menyimpan data pendanaan kedalam database. 14. Use case selesai.
<b>Alternative Flow</b>	: None
<b>Error Flow</b>	: E-1 Nominal dana yang dimasukkan salah. a. Sistem memberikan pesan peringatan bahwa nominal dana yang dimasukkan oleh aktor salah. b. Kembali ke Basic Flow langkah 8.

<b>Nama Use Case</b>	: Upload Bukti Bayar
<b>Deskripsi singkat</b>	: Usecase ini digunakan oleh aktor untuk melakukan upload bukti transaksi pendanaan proyek pertanian.
<b>Aktor</b>	: Member
<b>Prere-kondisi</b>	: 1. Aktor telah melakukan login sebagai member. 2. Aktor telah memasuki sistem.
<b>Post Kondisi</b>	: Data telah disimpan ke database.
<b>Basic Flow</b>	: 1. Use case ini dimulai ketika aktor memilih tab proses ku 2. Sistem menampilkan list proyek pertanian yang telah diproses beserta status prosesnya. 3. Aktor memilih salah satu proyek pertanian dengan status proses menunggu konfirmasi. 4. Sistem menampilkan form upload bukti transaksi. 5. Aktor memilih button Pilih Gambar...

	6. Sistem menampilkan pilihan file gambar dari penyimpanan 7. Aktor memilih salah satu gambar 8. Sistem menampilkan gambar yang dipilih pada form 9. Aktor memilih button Konfirmasi. 10. Sistem menampilkan list proyek pertanian dengan status sudah diproses. 11. Use case selesai.
Alternative Flow	: None
Error Flow	: None

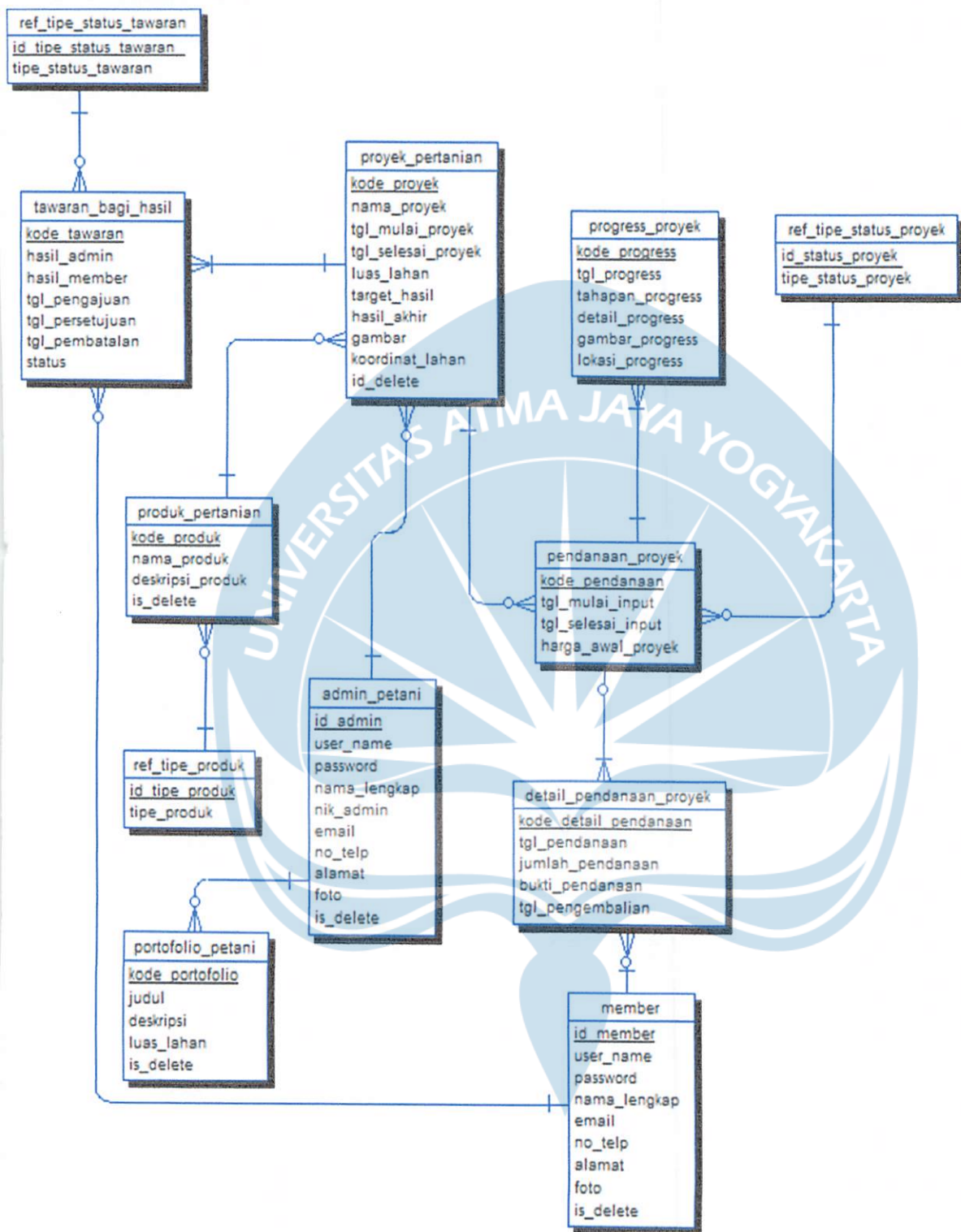
Nama Use Case	: Menawar Hasil Bagi (Member)
Deskripsi singkat	: Usecase ini digunakan oleh aktor untuk menawar maupun mengkonfirmasi tawaran yang masuk
Aktor	: Member
Pre kondisi	: <ol style="list-style-type: none"> <li>1. Aktor telah melakukan login sebagai petani.</li> <li>2. Aktor telah memasuki sistem.</li> </ol>
Pos Kondisi	: Data telah disimpan ke database.
Basic Flow	: <ol style="list-style-type: none"> <li>1. Use case ini dimulai ketika aktor memilih tab tawaran ku</li> <li>2. Sistem menampilkan list tawaran proyek pertanian yang telah diproses beserta status prosesnya.</li> <li>3. Aktor memilih salah satu item proyek pertanian dengan status proses menunggu.</li> <li>4. Sistem menampilkan persentase proyek pertanian tersebut.</li> <li>5. Aktor memilih button Setuju.</li> <li>A-1 Aktor memilih button Tidak Setuju</li> <li>6. Sistem menampilkan pesan tawaran disetujui.</li> <li>7. Use case selesai.</li> </ol>
Alternative Flow	:         A-1 Aktor memilih button Tidak Setuju <ol style="list-style-type: none"> <li>1. Sistem menampilkan form baru berisi persentase petani dan member</li> <li>2. Aktor mengisi form dengan persentase hasil bagi yang baru</li> <li>3. Sistem mengecek persentase hasil bagi valid atau tidak</li> <li>4. Aktor memilih button konfirmasi tawaran</li> <li>5. Sistem menampilkan list tawaran yang baru</li> <li>6. Belrlanjut ke basic flow langkah 7</li> </ol>
Error Flow	: None

Nama Use Case	: Tampil Laporan Progress Proyek Pertanian
Deskripsi singkat	: Usecase ini digunakan oleh aktor untuk melihat progress perkembangan proyek yang telah aktor danai.
Aktor	: Member
Pre kondisi	: <ol style="list-style-type: none"> <li>1. Aktor telah melakukan login sebagai Member.</li> <li>2. Aktor telah memasuki sistem.</li> </ol>

Pos Kondisi	:	Data progres proyek pertanian di database telah ter ditampilkan.
Basic Flow	:	<ol style="list-style-type: none"> <li>1. Use case ini dimulai ketika aktor memilih tab Show Progress.</li> <li>2. Sistem menampilkan daftar proyek pertanian yang telah didanai.</li> <li>3. Aktor memilih salah satu proyek pertanian.</li> <li>4. Sistem menampilkan progres proyek pertanian.</li> <li>5. Aktor memilih salah satu tahap progres proyek pertanian.</li> <li>6. Sistem menampilkan detail progress proyek pertanian tersebut.</li> <li>7. Use case selesai.</li> </ol>
Alternative Flow	:	None
Error Flow	:	None



### 3.4 Kebutuhan Data



Gambar 2 Entity Relationship Diagram (ERD)

**Pengembangan Aplikasi Mobile  
Investasi Pertanian  
MIP**

**Deskripsi Perancangan Perangkat Lunak**

**Esthy Yustiana Wahyu / 140707696**

**Dibuat untuk:  
Universitas Atma Jaya Yogyakarta  
2018**

## Daftar Revisi

Tanggal	Deskripsi Revisi	Penulis	Keterangan

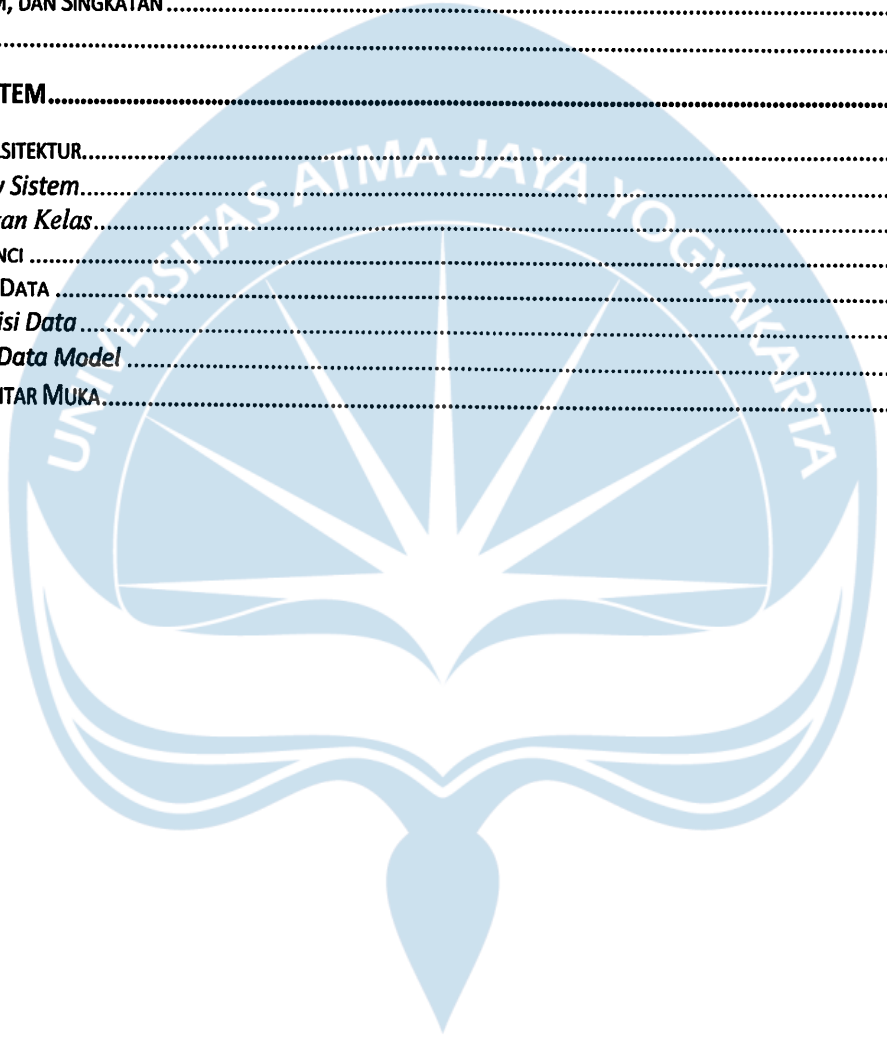
## Persetujuan Dokumen

Dokumen Deskripsi Perancangan Perangkat Lunak ini telah diterima dan disetujui oleh:

Nama	Tanggal	Tandatangan

# Daftar Isi

- DAFTAR REVISI ..... II
- PERSETUJUAN DOKUMEN ..... II
- 1. PENDAHULUAN ..... 1
  - 1.1 TUJUAN PENULISAN DOKUMEN ..... 1
  - 1.2 RUANG LINGKUP ..... 1
  - 1.3 DEFINISI, AKRONIM, DAN SINGKATAN ..... 2
  - 1.4 REFERENSI ..... 2
- 2. PERANCANGAN SISTEM ..... 3
  - 2.1 PERANCANGAN ARSITEKTUR ..... 3
    - 2.1.1. *Overview Sistem* ..... 3
    - 2.1.2 *Perancangan Kelas* ..... 4
  - 2.2 PERANCANGAN RINCI ..... 6
  - 2.3 PERANCANGAN DATA ..... 16
    - 2.3.1 *Dekomposisi Data* ..... 16
    - 2.3.2 *Physical Data Model* ..... 22
  - 2.4 PERANCANGAN ANTAR MUKA ..... 23





# 1. Pendahuluan

## 1.1 Tujuan Penulisan Dokumen

Dokumen Deskripsi Perancangan Perangkat Lunak (DPPL) ini merupakan dokumen deskripsi perancangan perangkat lunak aplikasi mobile investasi pertanian untuk memberikan deskripsi rancangan semua fungsionalitas, kebutuhan data, antar muka, karakteristik atau atribut yang dimiliki oleh aplikasi.

Dokumen Deskripsi Perancangan Perangkat Lunak (DPPL) ini ditulis untuk pengembang selanjutnya, yang akan melanjutkan pengembangan aplikasi investasi pertanian ini.

## 1.2 Ruang Lingkup

Ruang lingkup perangkat lunak ini dikembangkan sebagai berikut:

Perangkat lunak investasi pertanian ini dibangun dengan menggunakan *tools* Android Studio dengan menggunakan database MySQL. Perangkat lunak ini akan berjalan pada lingkungan mobile dengan sistem operasi Android. Perangkat lunak akan berkomunikasi dengan database menggunakan API.

Perangkat lunak investasi pertanian ini dibangun dengan fungsi yaitu pengelolaan proyek pertanian, baik menambah data baru, mengubah data, dan menghapusnya. Fungsi lainnya adalah pengelolaan proses pertanian. Fungsi utama dari aplikasi ini adalah fungsi tawar-menawar dan fungsi pendanaan pada suatu proyek. Perangkat lunak ini tidak dapat menangani transaksi pembayaran secara online, juga tidak menangani proses akhir pengembalian keuntungan dari petani ke investornya.

Perangkat Lunak ini dikembangkan dengan sasaran guna membantu petani Indonesia untuk mendapatkan modal lebih mudah dengan cara membuka peluang investasi untuk produk pertanian maupun peternakan mereka.

### 1.3 Definisi, Akronim, dan Singkatan

Definisi	
Investor	Orang perorangan atau lembaga baik domestik atau non domestik yang melakukan suatu investasi (bentuk penanaman modal sesuai dengan jenis investasi yang dipilihnya) baik dalam jangka pendek atau jangka panjang.
Petani	Seseorang yang bergerak di bidang pertanian, dengan cara melakukan pengelolaan tanah dengan tujuan untuk menumbuhkan dan memelihara tanaman, dengan harapan untuk memperoleh hasil.
Tawar-Menawar	Proses dimana dua atau lebih pihak berusaha untuk menyelesaikan apa yang masing-masing akan beri dan akan menerima, atau yang masing-masing lakukan dan terima dalam suatu transaksi diantara mereka

Akronim dan Singkatan	
DPPL	Deskripsi Perancangan Perangkat Lunak
MIP	Mobile Investasi Pertanian
DPPL-MIP-XXX-YY	Kode yang merepresentasikan kebutuhan pada aplikasi dimana XXX merupakan nomor fungsi produk dan YY merupakan subfungsi produk.

### 1.4 Referensi

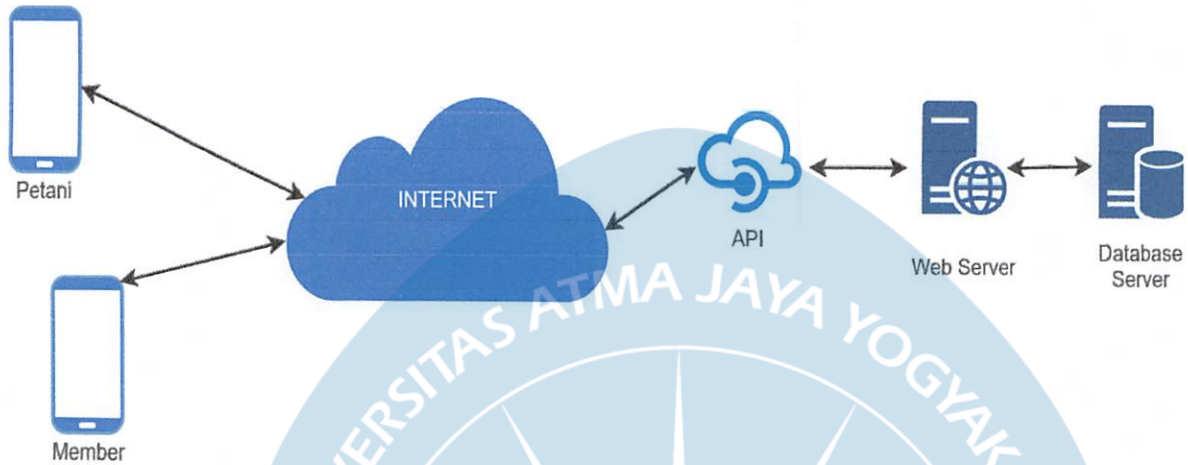
Referensi yang digunakan pada perangkat lunak yang dikembangkan adalah berikut:

Wahyu, Esthy Yustiana (2016). *DESKRIPSI PERANCANGAN PERANGKAT LUNAK SNBC*. Yogyakarta: Universitas Atma Jaya Yogyakarta.

## 2. Perancangan Sistem

### 2.1 Perancangan Arsitektur

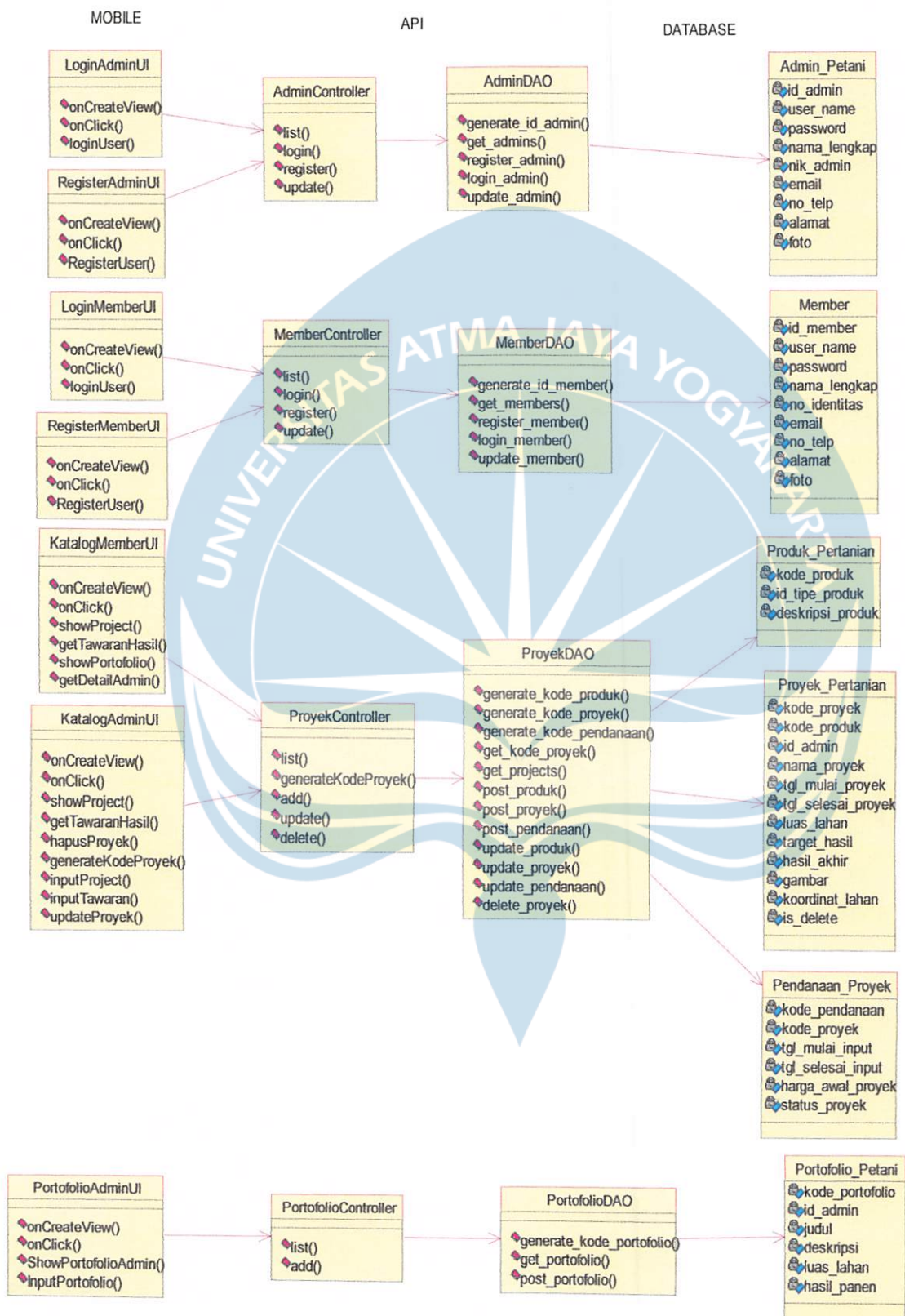
#### 2.1.1. Overview Sistem



Gambar 1 Arsitektur Sistem Aplikasi Investasi

Gambar diatas merupakan arsitektur dari Perangkat Lunak Investasi Pertanian Mobile. Aplikasi dijalankan pada perangkat keras mobile dengan sistem operasi android. Arsitektur perangkat lunak investasi pertanian menggunakan model *client-server* dimana administrator dan member merupakan *client*. Admisitrator dan member dapat berkomunikasi dengan database menggunakan API yang telah dibuat. *Client* mengirim *request* ke *server* menggunakan API yang kemudian API meneruskannya ke database dan akan dikirimkan kembali datanya dalam bentuk JSON. JSON ini nantinya yang dibaca oleh aplikasi dan membuat *client* mendapatkan *response* dari *server*.

## 2.1.2 Perancangan Kelas

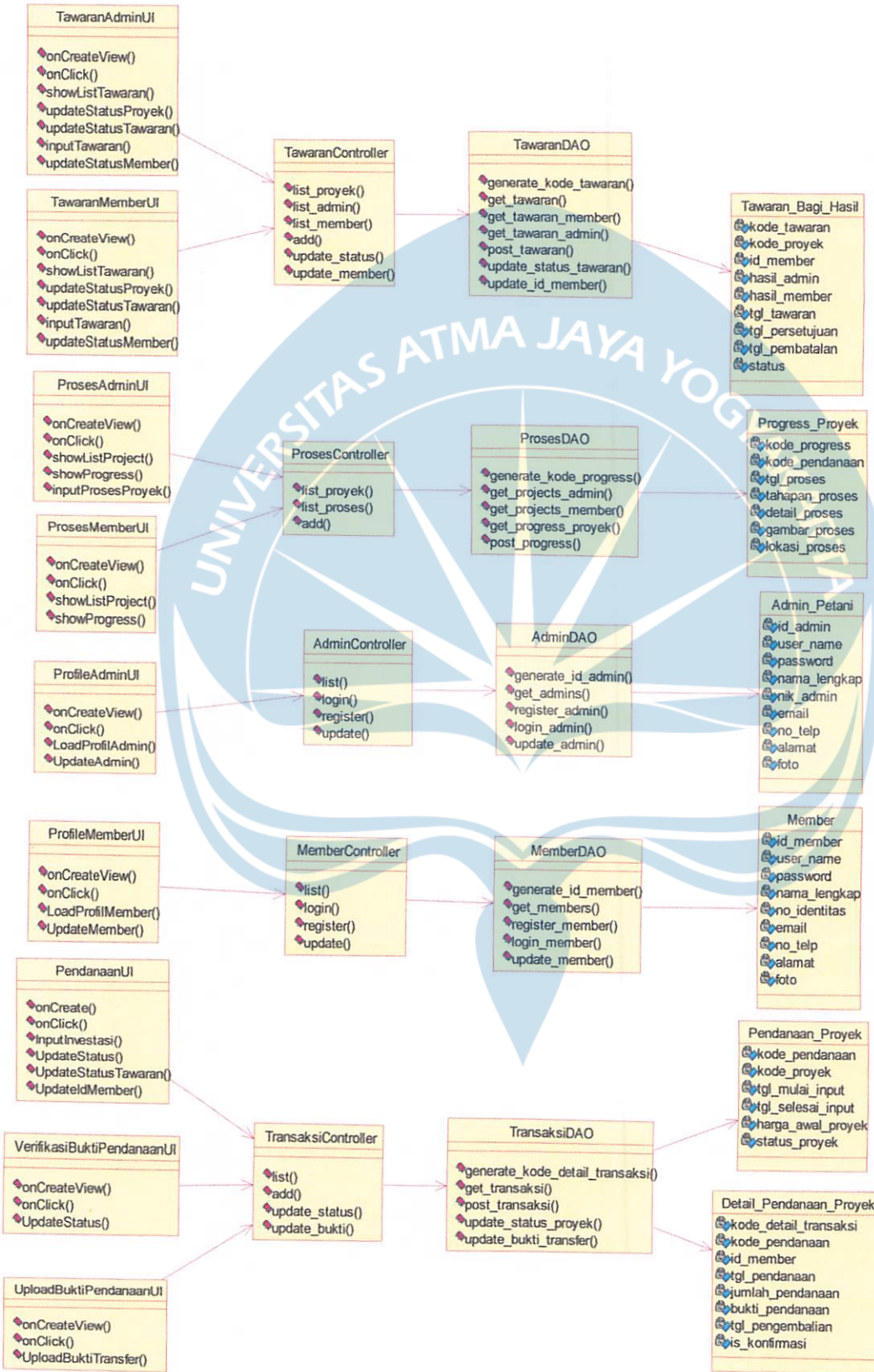


Gambar 2 Kelas Diagram Aplikasi

MOBILE

API

DATABASE



Gambar 3 Kelas Diagram Aplikasi

## 2 Perancangan Rinci

### Deskripsi atribut kelas Admin\_Petani

Nama atribut	Tipe Data	Deskripsi
Id_admin	Char (10)	Digenerate oleh sistem
User_name	Variable char (50)	Harus unik
Password	Variable char (100)	Format SHA-1 yang sudah digenerate sistem
Nama_lengkap	Variable char (100)	Nama pengguna
Nik_admin	Variable char (16)	Nomor Induk Kependudukan
Email	Variable char (50)	Email pengguna
No_telp	Variable char (25)	No telp / handphone pengguna
Alamat	Variable char (100)	Alamat pengguna
Foto	Variable char (100)	Merupakan direktori dari foto
Is_delete	Char (10)	Status aktif pengguna

### deskripsi fungsi pada kelas Admin\_Petani

generate_id_admin()	
Input	-
Output	Id_admin
Deskripsi	Id_admin menggunakan format ADMXXXX, X merupakan no urut admin, misal ADM0001, ADM0002

get_admins(string id_admin)	
Input	Id_admin
Output	Data admin (id_admin, user_name, password, nama_lengkap, nik_admin, email, no_telp, alamat, foto, is_delete)
Deskripsi	Proses mendapatkan data lengkap admin

register_admin (string user_name, string password, string nama_lengkap, string nik_admin, string email, string id_admin)	
Input	user_name, password, nama_lengkap, nik_admin, email, id_admin
Output	Data pengguna telah disimpan dalam database
Deskripsi	Proses memasukkan data pengguna ke database, id_admin digenerate dengan fungsi generate_id_admin(), dan password di enkripsi menggunakan SHA-1

login_admin (string user_name, string password)	
Input	user_name, password
Output	Data admin (id_admin, user_name, password, nama_lengkap, nik_admin, email, no_telp, alamat, foto, is_delete)
Deskripsi	Proses masuk ke dalam aplikasi.

<b>update_admin (string user_name, string nama_lengkap, string nik_admin, string email, string no_telp, string alamat, string foto, string id_admin)</b>	
Input	user_name, nama_lengkap, nik_admin, email, no_telp, alamat, foto, id_admin
Output	Data pengguna telah diupdate dalam database
Deskripsi	Proses mengubah data pengguna berdasarkan id_admin pengguna.

**Deskripsi atribut kelas Member**

Nama atribut	Tipe Data	Deskripsi
Id_member	Char (10)	Digenerate oleh sistem
User_name	Variable char (50)	Harus unik
Password	Variable char (100)	Format SHA-1 yang sudah digenerate sistem
Nama_lengkap	Variable char (100)	Nama pengguna
No_identitas	Variable char (16)	Nomor Induk Kependudukan
Email	Variable char (50)	Email pengguna
No_telp	Variable char (25)	No telp / handphone pengguna
Alamat	Variable char (100)	Alamat pengguna
Foto	Variable char (100)	Merupakan direktori dari foto
Is_delete	Char (10)	Status aktif pengguna

**eskripsi fungsi pada kelas Member**

<b>generate_id_member()</b>	
Input	-
Output	Id_member
Deskripsi	Id_member menggunakan format YYYYMMDDXXXX, YYYY merupakan tahun, MM merupakan bulan, DD merupakan tanggal, XXXX merupakan no urut pengguna

<b>get_members(string id_member)</b>	
Input	Id_member
Output	Data member (id_member, user_name, password, nama_lengkap, no_identitas, email, no_telp, alamat, foto, is_delete)
Deskripsi	Proses mendapatkan data lengkap member

<b>register_member(string user_name, string password, string nama_lengkap, string no_identitas, string email, string id_member)</b>	
Input	user_name, password, nama_lengkap, no_identitas, email, id_member
Output	Data pengguna telah disimpan dalam database
Deskripsi	Proses memasukkan data pengguna ke database, id_member digenerate dengan fungsi generate_id_member(), dan password di enkripsi menggunakan SHA-1

<b>login_member(string user_name, string password)</b>	
Input	user_name, password
Output	Data member (id_member, user_name, password, nama_lengkap, no_identitas, email, no_telp, alamat, foto, is_delete)
Deskripsi	Proses masuk ke dalam aplikasi.

<b>update_member (string user_name, string nama_lengkap, string nik_admin, string email, string no_telp, string alamat, string foto, string id_member)</b>	
Input	user_name, nama_lengkap, no_identitas, email, no_telp, alamat, foto, id_member
Output	Data pengguna telah diupdate dalam database
Deskripsi	Proses mengubah data pengguna berdasarkan id_member pengguna.

#### Deskripsi atribut kelas Ref\_tipe\_produk

Nama atribut	Tipe Data	Deskripsi
id_tipe_produk	Char (10)	No urut tipe produk
Tipe_produk	Variable char (50)	Tipe / jenis produk

#### Deskripsi atribut kelas Ref\_tipe\_status\_proyek

Nama atribut	Tipe Data	Deskripsi
id_tipe_status_proyek	Char (10)	No urut tipe status proyek
Tipe_status_proyek	Variable char (50)	Tipe / jenis status proyek

#### Deskripsi atribut kelas Ref\_tipe\_status\_tawaran

Nama atribut	Tipe Data	Deskripsi
id_tipe_status_tawaran	Char (10)	No urut tipe status tawaran
Tipe_status_tawaran	Variable char (50)	Tipe / jenis status tawaran

#### Deskripsi atribut kelas Produk\_pertanian

Nama atribut	Tipe Data	Deskripsi
Code_produk	Variable char (25)	Digenerate oleh sistem
id_tipe_produk	Char (10)	Sesuai dengan tabel ref_tipe_produk
Nama_produk	Variable char (50)	Nama produk
Deskripsi_produk	Variable char (100)	Deskripsi produk
is_delete	Char (10)	Status produk dihapus



deskripsi fungsi pada kelas **Produk\_pertanian**

<b>generate_kode_produk(string id_tipe_produk)</b>	
Input	Id_tipe_produk
Output	Kode_produk
Deskripsi	Kode_produk menggunakan format Y-XXXX, Y merupakan tipe produk dan XXXX merupakan no urut produk

<b>post_produk (string id_tipe_produk, string nama_produk, string deskripsi_produk, string kode_produk )</b>	
Input	Id_tipe_produk, nama_produk, deskripsi_produk, kode_produk
Output	Data produk telah disimpan pada database
Deskripsi	Proses menyimpan data produk baru. Kode_produk didapatkan dengan fungsi generate_kode_produk(id_tipe_produk)

<b>update_produk (string id_tipe_produk, string nama_produk, string deskripsi_produk, string kode_produk )</b>	
Input	Id_tipe_produk, nama_produk, deskripsi_produk, kode_produk
Output	Data produk telah disimpan pada database
Deskripsi	Proses menyimpan data produk baru. Kode_produk didapatkan dengan fungsi generate_kode_produk(id_tipe_produk)

deskripsi atribut kelas **Proyek\_pertanian**

Nama atribut	Tipe Data	Deskripsi
kode_proyek	Variable char (25)	Digenerate oleh sistem
kode_produk	Variable char (25)	Produk dari proyek
id_admin	Char (10)	Petani yang menambahkan / memiliki proyek
nama_proyek	Variable char (50)	Judul proyek agar menarik investor
tgl_mulai_proyek	Date	Tanggal mulai proyek berlangsung
tgl_selesai_proyek	Date	Tanggal proyek selesai / tanggal selesai kontrak
luas_lahan	Variable char (100)	Luas lahan proyek
target_hasil	Char (10)	Target hasil proyek
hasil_akhir	Char (10)	Hasil akhir jika proyek telah selesai
gambar	Variable char (100)	Gambar yang mendeskripsikan proyek, dapat merupakan gambar produk
koordinat_lahan	Variable char (100)	Lokasi dimana proyek dilakukan
status_delete	Int (10)	Status dari proyek

Deskripsi fungsi pada kelas **Proyek\_pertanian**

<b>generate_kode_proyek()</b>	
Input	-
Output	Kode_proyek
Deskripsi	Kode_proyek menggunakan format PRO-YYYYMMDD-XXXX, YYYYMMDD merupakan tahun bulan tanggal dibuat proyek, XXXX merupakan nomor urut proyek

<b>Getkodeproyek()</b>	
Input	-
Output	Json Kode_proyek
Deskripsi	Kode_proyek didapatkan dari fungsi generate_kode_proyek(), kemudian pada fungsi ini kode_proyek tersebut dikirim dalam bentuk json ke aplikasi

<b>Get_projects(string id_admin)</b>	
Input	Id_admin
Output	Data proyek ( kode_pendanaan, kode_proyek, kode_produk, id_admin, nama_proyek, tgl_mulai_proyek, tgl_selesai_proyek, luas_lahan, target_hasil, hasil_akhir, foto, koordinat_lahan, harga_awal_proyek, status_proyek, nama_produk, tipe_produk, tipe_status_proyek, id_tipe_proyek)
Deskripsi	Proses mendapatkan data proyek berdasarkan id_admin tertentu, maupun data proyek semua yang tersimpan pada database

<b>Post_proyek (string kode_proyek, string id_admin, string nama_proyek, date tgl_mulai_proyek, date tgl_selesai_proyek, string luas_lahan, string target_hasil, string gambar, string koordinat_lahan)</b>	
Input	kode_proyek, id_admin, nama_proyek, tgl_mulai_proyek, tgl_selesai_proyek, luas_lahan, target_hasil, gambar, koordinat_lahan
Output	Data proyek baru telah tersimpan di proyek_pertanian
Deskripsi	Proses memasukkan data proyek baru kedalam database

<b>update_proyek (string kode_proyek, string nama_proyek, date tgl_mulai_proyek, date tgl_selesai_proyek, string luas_lahan, string target_hasil, string gambar, string koordinat_lahan)</b>	
Input	kode_proyek, nama_proyek, tgl_mulai_proyek, tgl_selesai_proyek, luas_lahan, target_hasil, gambar, koordinat_lahan
Output	Data proyek telah tersimpan di proyek_pertanian
Deskripsi	Proses mengubah data proyek pada database

<b>delete_proyek (string kode_proyek)</b>	
Input	kode_proyek
Output	Status is_delete telah berubah menjadi 1
Deskripsi	Proses menghapus data proyek pada database

Deskripsi atribut kelas **Pendanaan\_proyek**

Nama atribut	Tipe Data	Deskripsi
kode_pendanaan	Variable char (25)	Digenerate oleh sistem
kode_proyek	Variable char (25)	Kode_proyek yang akan didanai
tgl_mulai_input	Timestamp	Tanggal input proyek
tgl_selesai_input	Timestamp	Batas tanggal input proyek
harga_awal_proyek	Char (15)	Harga / jumlah dana yang dibutuhkan proyek
status_proyek	Char (10)	Status proyek (belum didanai, menunggu konfirmasi, telah didanai, menunggu tawaran)

Deskripsi fungsi pada kelas **Penadanaan\_pertanian**

generate_kode_pendanaan()	
Input	-
Output	Kode_pendanaan
Deskripsi	Kode_pendanaan menggunakan format Dna-YYYYMMDD-XXXX, YYYYMMDD merupakan tahun bulan tanggal dibuat proyek, XXXX merupakan nomor urut pendanaan

tambah_pendanaan(string kode_proyek, timestamp tgl_mulai_input, timestamp tgl_selesai_input, char harga_awal_proyek, string kode_pendanaan)	
Input	kode_proyek, tgl_mulai_input, tgl_selesai_input, harga_awal_proyek, kode_pendanaan
Output	Data pendanaan baru telah disimpan
Deskripsi	Proses memasukkan data pendanaan baru ke dalam pendanaan_proyek. Kode_pendanaan didapatkan dengan fungsi generate_kode_pendanaan().

update_pendanaan(string kode_pendanaan, char harga_awal_proyek)	
Input	harga_awal_proyek, kode_pendanaan
Output	Data pendanaan telah diubah
Deskripsi	Proses mengubah data pendanaan berdasarkan kode_pendanaan tertentu

update_status_proyek(string kode_pendanaan, string status_proyek)	
Input	Kode_pendanaan, status_proyek
Output	Status proyek berubah menjadi status proyek yang baru
Deskripsi	Proses mengubah status proyek dengan status baru berdasarkan tabel ref_tipe_status_proyek

<b>Get_projects_admin( string id_admin)</b>	
Input	Id_admin
Output	Data proyek yang telah didanai member berdasarkan id_admin (kode_proyek, kode_produk, kode_pendanaan, id_admin, id_member, nama_proyek, nama_produk, tgl_mulai_proyek, tgl_selesai_proyek, gambar, harga_awal_proyek, status_proyek, nama_lengkap, jumlah_pendanaan, tipe_status_proyek )
Deskripsi	Proses mendapatkan proyek pada petani tertentu yang status proyeknya telah berubah (telah didanai, menunggu konfirmasi, maupun menunggu tawaran)

<b>Get_projects_member( string id_member)</b>	
Input	Id_member
Output	Data proyek yang telah didanai member, maupun yang sedang menunggu konfirmasi (kode_proyek, kode_produk, kode_pendanaan, id_admin, id_member, nama_proyek, nama_produk, tgl_mulai_proyek, tgl_selesai_proyek, gambar, harga_awal_proyek, status_proyek, nama_lengkap, jumlah_pendanaan, tipe_status_proyek, kode_detail_transaksi)
Deskripsi	Proses mendapatkan proyek pada member tertentu yang status proyeknya telah berubah (telah didanai, menunggu konfirmasi, maupun menunggu tawaran)

**Deskripsi atribut kelas detail\_pendanaan**

Nama atribut	Tipe Data	Deskripsi
Kode_detail_transaksi	Variable char (25)	Digenerate oleh sistem
Kode_pendanaan	Variable char (25)	Kode_pendanaan proyek yang telah dipilih member
Id_member	Char (15)	Id member proyek
Tgl_pendanaan	Timestamp	Tgl pendanaan / tgl transaksi
Jumlah_pendanaan	Char (15)	Jumlah dana yang diberikan member
Bukti_pendanaan	Variable char (100)	Bukti pendanaan berupa foto bukti transfer
Tgl_pengembalian	Timestamp	Tgl pengembalian dana
Is_konfirmasi	Int (1)	Status transaksi sudah dikonfirmasi petani

**Deskripsi fungsi pada kelas Detail\_pendanaan**

<b>generate_kode_detail_transaksi()</b>	
Input	-
Output	Kode_detail_transaksi
Deskripsi	Kode_detail_transaksi menggunakan format DTL-YYYYMMDD-XXXX, YYYYMMDD merupakan tahun bulan tanggal dibuat proyek, XXXX merupakan nomor urut transaksi

<b>Get_transaksi(string kode_pendanaan)</b>	
Input	Kode_pendanaan
Output	Data detail pendanaan (kode_pendanaan, id_member, nama_member, no_telp_member, tgl_pendanaan, jumlah_pendanaan, bukti_pendanaan, tgl_pengembalian, is_konfirmasi)
Deskripsi	Proses mendapatkan data detail pendanaan / data transaksi

<b>post_transaksi(string kode_pendanaan, string id_member, char jumlah_pendanaan, date tgl_pengembalian, date tgl_pendanaan, string kode_detail_transaksi)</b>	
Input	kode_pendanaan, id_member, jumlah_pendanaan, tgl_pengembalian, tgl_pendanaan, kode_detail_transaksi
Output	Data detail pendanaan baru telah tersimpan pada database
Deskripsi	Proses memasukkan data detail pendanaan baru ke tabel detail_pendanaan. Kode_detail_transaksi digenerate dengan fungsi generate_kode_detail_transaksi

<b>Update_bukti_transfer(string kode_detail_transaksi, string bukti_pendanaan)</b>	
Input	Kode_detail_pendanaan, bukti_pendanaan
Output	Bukti pendanaan terupdate
Deskripsi	Proses mengupdate kolom bukti pendanaan

10. Deskripsi atribut kelas **progress\_proyek**

Nama atribut	Tipe Data	Deskripsi
Kode_progress	Variable char (25)	Digenerate oleh sistem
Kode_pendanaan	Variable char (25)	Kode pendanaan pada proses proyek
Tgl_progress	Date	Tanggal proses diinputkan
Tahapan_progress	Variable char (100)	Tahapan proses proyek
Detail_progress	Variable char (100)	Detail proses proyek yang diinputkan
Gambar_progress	Variable char (100)	Gambar terkini dari proses proyek
Lokasi_progress	Variable char (100)	Lokasi proses yang dilakukan

Deskripsi fungsi pada kelas **Progress\_proyek**

<b>generate_kode_progress()</b>	
Input	-
Output	Kode_progress
Deskripsi	Kode_progress menggunakan format Prg-YYYYMMDD-XXXX, YYYYMMDD merupakan tahun bulan tanggal dibuat progress, XXXX merupakan nomor urut progress

<b>get_progress_proyek(string kode_pendanaan)</b>	
Output	Kode_pendanaan
Input	Data proses proyek (kode_progress, kode_pendanaan, tgl_progress, detail_progress, gambar_progress, lokasi_progress)
Deskripsi	Proses mendapatkan data perkembangan / proses proyek berdasarkan kode_pendanaannya

<b>save_progress(string kode_pendanaan, string kode_progress, date tgl_progress, string tahapan_progress, string detail_progress, string gambar_progress, string lokasi_progress)</b>	
Output	kode_pendanaan, kode_progress, tgl_progress, tahapan_progress, detail_progress, gambar_progress, lokasi_progress
Input	Data proses proyek baru telah tersimpan didatabase
Deskripsi	Proses memasukkan data perkembangan proyek baru ke database. Kode_progress didapat menggunakan fungsi generate_kode_progress()

**Deskripsi atribut kelas portofolio\_petani**

Nama atribut	Tipe Data	Deskripsi
kode_portofolio	Variable char (25)	Digenerate oleh sistem
id_admin	Char (10)	Id_admin dari pengguna
judul	Variable char (50)	Judul portofolio
deskripsi	Variable char (100)	Deskripsi portofolio
luas_lahan	Variable char (100)	Luas lahan yang pernah dikerjakan
hasil_panen	double	Hasil panen dari proyek sebelumnya
delete	Char (10)	Status portofolio

**Deskripsi fungsi pada kelas Potofolio\_petani**

<b>generate_kode_portofolio()</b>	
Output	-
Input	Kode_portofolio
Deskripsi	Kode_portofolio menggunakan format PTF-YYYYMMDD-XXXX, YYYYMMDD merupakan tahun bulan tanggal dibuat portofolio, XXXX merupakan nomor urut portofolio

<b>get_portofolio(string id_admin)</b>	
Output	Id_admin
Input	Data portofolio (kode_portofolio, id_admin, judul, deskripsi, luas_lahan, hasil_panen, is_delete)
Deskripsi	Proses mendapatkan data portofolio untuk petani tertentu

<b>save_portofolio(string id_admin, string kode_portofolio, string judul, string deskripsi, string luas_lahan, double hasil_panen)</b>	
Output	id_admin, kode_portofolio, judul, deskripsi, luas_lahan, hasil_panen
Input	Data portofolio baru telah tersimpan didatabase

Deskripsi	Proses memasukkan portofolio petani ke database. Kode_portofolio didapatkan menggunakan fungsi generate_kode_portofolio()
-----------	---

## 12. Deskripsi atribut kelas tawaran\_bagi\_hasil

Nama atribut	Tipe Data	Deskripsi
Kode_tawaran	Variable char (25)	Digenerate oleh sistem
Kode_proyek	Variable char (25)	Kode proyek dari tawaran proyek
Id_member	Char (15)	Id member dari member yang mengajukan tawaran
Hasil_admin	Char (10)	Persentase hasil petani
Hasil_member	Char (10)	Persentase hasil member
Tgl_tawaran	date	Tanggal tawaran dibuat
Tgl_persetujuan	date	Tanggal proyek disetujui
Tgl_pembatalan	date	Tanggal proyek tidak disetujui
status	Int (10)	Status dari sebuah tawaran

## Deskripsi fungsi pada kelas tawaran\_bagi\_hasil

generate_kode_tawaran()	
Input	-
Output	Kode_tawaran
Deskripsi	Kode_tawaran menggunakan format TWR-YYYYMMDD-XXXX, YYYYMMDD merupakan tahun bulan tanggal dibuat tawaran, XXXX merupakan nomor urut tawaran

Get_tawaran(string kode_proyek)	
Input	Kode_proyek
Output	Data tawaran proyek tertentu (kode_tawaran, kode_proyek, hasil_admin, hasil_member, tgl_tawaran, tgl_persetujuan, tgl_pembatalan, status_tawaran, tipe_status_tawaran, id_member, nama_proyek, kode_pendanaan, harga_proyek, tgl_selesai_proyek)
Deskripsi	Proses mendapatkan data tawaran pada proyek tertentu dengan menggunakan kode_proyek.

Get_tawaran_member(string id_member)	
Input	Id_member
Output	Data tawaran member tertentu (kode_tawaran, kode_proyek, hasil_admin, hasil_member, tgl_tawaran, tgl_persetujuan, tgl_pembatalan, status_tawaran, tipe_status_tawaran, id_member, nama_proyek, kode_pendanaan, harga_proyek, tgl_selesai_proyek)
Deskripsi	Proses mendapatkan data tawaran pada member tertentu dengan menggunakan id_member.

<b>Get_tawaran_admin(string id_admin)</b>	
Input	Id_admin
Output	Data tawaran petani tertentu (kode_tawaran, kode_proyek, hasil_admin, hasil_member, tgl_tawaran, tgl_persetujuan, tgl_pembatalan, status_tawaran, tipe_status_tawaran, id_member, nama_proyek, kode_pendanaan, harga_proyek, tgl_selesai_proyek)
Deskripsi	Proses mendapatkan data tawaran pada petani tertentu dengan menggunakan id_admin.

<b>Post_tawaran (string kode_proyek, string id_member, string hasil_admin, string hasil_member, date tgl_tawaran, string kode_tawaran)</b>	
Input	kode_proyek, id_member, hasil_admin, hasil_member, tgl_tawaran, kode_tawaran
Output	Data tawaran baru telah tersimpan dalam database
Deskripsi	Proses memasukkan data tawaran baru kedalam database. Kode_tawaran digenerate menggunakan fungsi generate_kode_tawaran

<b>Update_status_tawaran(string kode_tawaran, string id_status, date tgl_persetujuan, date tgl_pembatalan)</b>	
Input	kode_tawaran, id_status, tgl_persetujuan, tgl_pembatalan
Output	Data tawaran diupdate. Kolom yang diupdate adalah kolom status_tawaran, tgl_persetujuan, dan tgl_pembatalan
Deskripsi	Proses mengupdate data status tawaran dan mengupdate tanggal persetujuan dan tanggal pembatalan

<b>Update_id_member(string kode_tawaran, string id_member)</b>	
Input	kode_tawaran, id_member
Output	Data tawaran diupdate. Kolom yang diupdate adalah kolom id_member
Deskripsi	Proses mengupdate data kolom id_member

## 2.3 Perancangan Data

### 2.3.1 Dekomposisi Data

Fabel 1 : Struktur Tabel Admin\_petani

Nama Field	Tipe Data	Null	Konstrain	Range Nilai	Default	Keterangan
Id_admin	CHAR(10)	NO		10		PRIMARY KEY
User_name	VARCHAR(50)	NO	Harus unik	50		
Password	VARCHAR(100)	NO	Format SHA-1	100		
Nama_lengkap	VARCHAR(100)	NO		100		
Nik_admin	VARCHAR(16)	NO	Format angka 0 - 9	16		



Nama Field	Tipe Data	Null	Konstrain	Range Nilai	Default	Keterangan
Email	VARCHAR(50)	NO	Format email	50		
No_telp	VARCHAR(25)	YES	Format angka 0 – 9	25		
Alamat	VARCHAR(100)	YES		100		
Foto	VARCHAR(100)	YES		100		
Is_delete	CHAR(10)	NO	Format angka 0 - 1	10		

Tabel 2 : Struktur Tabel Member

Nama Field	Tipe Data	Null	Konstrain	Range Nilai	Default	Keterangan
Id_member	CHAR(15)	NO		10		PRIMARY KEY
User_name	VARCHAR(50)	NO	Harus unik	50		
Password	VARCHAR(100)	NO	Format SHA-1	100		
Nama_lengkap	VARCHAR(100)	NO		100		
No_identitas	INT(20)	NO	Format angka 0 - 9	16		
Email	VARCHAR(50)	NO	Format email	50		
No_telp	VARCHAR(25)	YES	Format angka 0 – 9	25		
Alamat	VARCHAR(100)	YES		100		
Foto	VARCHAR(100)	YES		100		
Is_delete	CHAR(10)	NO	Format angka 0 - 1	10	0	

Tabel 3 : Struktur Tabel Ref\_tipe\_produk

Nama Field	Tipe Data	Null	Konstrain	Range Nilai	Default	Keterangan
Id_tipe_produk	Char (10)	NO		1 - 10		PRIMARY KEY
Tipe_produk	Varchar (30)	YES		50		

Tabel 4 : Struktur Tabel Ref\_tipe\_status\_proyek

Nama Field	Tipe Data	Null	Konstrain	Range Nilai	Default	Keterangan
Id_tipe_status_proyek	Char (10)	NO		1 - 10		PRIMARY KEY
Tipe_status_proyek	Varchar (30)	YES		30		

Tabel 5 : Struktur Tabel Ref\_tipe\_status\_tawaran

Nama Field	Tipe Data	Null	Konstrain	Range Nilai	Default	Keterangan
Id_tipe_status_tawaran	Char (10)	NO		1 - 10		PRIMARY KEY
Tipe_status_tawaran	Varchar (30)	YES		30		

Tabel 6 : Struktur Tabel Produk\_pertanian

Nama Field	Tipe Data	Null	Konstrain	Range Nilai	Default	Keterangan
Kode_produk	Varchar (25)	NO	Generate sistem	25		PRIMARY KEY
Id_tipe_produk	Char (10)	NO		10		FOREIGN KEY dari tabel Ref_tipe_produk
Nama_produk	Varchar (50)	YES		50		
Deskripsi_produk	Varchar (100)	YES		100		
Is_delete	Char (10)	NO	Format angka 0 - 1	10	0	

Tabel 7 : Struktur Tabel Proyek\_pertanian

Nama Field	Tipe Data	Null	Konstrain	Range Nilai	Default	Keterangan
Kode_proyek	Varchar (25)	NO	Generate sistem	25		PRIMARY KEY
Kode_produk	Varchar (25)	NO		25		FOREIGN KEY dari tabel produk_pertanian
Id_admin	Char (10)	NO		10		FOREIGN KEY dari tabel admin_pertanian
Nama_proyek	Varchar (50)	NO		50		
Tgl_mulai_proyek	Date	NO	Tidak boleh melewati SYSDATE		0000-00-00	
Tgl_selesai_proyek	Date	NO	Tidak boleh kurang dari tanggal mulai proyek		0000-00-00	
Luas_lahan	Varchar (100)	NO	Format angka 0 - 9	100		
Target_hasil	Char (10)	NO	Format angka 0 - 9	10		

Nama Field	Tipe Data	Null	Konstrain	Range Nilai	Default	Keterangan
Hasil_akhir	Char (10)	YES	Format angka 0 - 9	10		
Gambar	Varchar (100)	NO		100		
Koordinat_lahan	Varchar (100)	YES		100		
Is_delete	Int (10)	NO	Format angka 0 - 1	10		

Tabel 8 : Struktur Tabel pendanaan\_proyek

Nama Field	Tipe Data	Null	Konstrain	Range Nilai	Default	Keterangan
Kode_pendanaan	Varchar (25)	NO	Generate sistem	25		PRIMARY KEY
Kode_proyek	Varchar (25)	NO		25		FOREIGN KEY dari tabel proyek_pertanian
Tgl_mulai_input	Timestamp	NO	SYSDATE		0000-00-00 00:00:00	
Tgl_selesai_input	Timestamp	NO	Tidak boleh kurang dari tanggal mulai input		0000-00-00 00:00:00	
Harga_awal_proyek	Char (15)	YES	Format angka desimal 0 - 1	1-15		
Status_proyek	Char (10)	NO		1-10	1	FOREIGN KEY dari tabel ref_tipe_status_proyek

Tabel 9 : Struktur Tabel detail\_pendanaan

Nama Field	Tipe Data	Null	Konstrain	Range Nilai	Default	Keterangan
Kode_detail_transaksi	Varchar (25)	NO	Generate sistem	25		PRIMARY KEY
Kode_pendanaan	Varchar (25)	NO		25		FOREIGN KEY dari tabel proyek_pertanian

Nama Field	Tipe Data	Null	Konstrain	Range Nilai	Default	Keterangan
member	Char (15)	NO		15		FOREIGN KEY dari tabel member
pendanaan	Timestamp	NO	SYSDATE		0000-00-00 00:00:00	
lah_pendanaan	Char (15)	NO	Format angka desimal 0-9	1 -15		Tidak boleh kurang dari harga awal proyek
ti_pendanaan	Varchar (100)	YES		1 - 100		
pengembalian	Timestap	YES	Tidak boleh kurang dari tanggal pendanaan		0000-00-00 00:00:00	
onfirmasi	Int (1)	NO		1	0	

#### 10 : Struktur Tabel progress\_proyek

Nama Field	Tipe Data	Null	Konstrain	Range Nilai	Default	Keterangan
le_progress	Varchar (25)	NO	Generate sistem	25		PRIMARY KEY
le_pendanaan	Varchar (25)	NO		25		FOREIGN KEY dari tabel proyek_pertanian
_progress	Date	NO	SYSDATE		0000-00-00	
apan_progress	Varchar (100)	YES		1 - 100		
ail_progress	Varchar (100)	YES		1 - 100		
mbar_progress	Varchar (100)	YES		1 - 100		
asi_progress	Varchar (100)	YES		1 - 100		

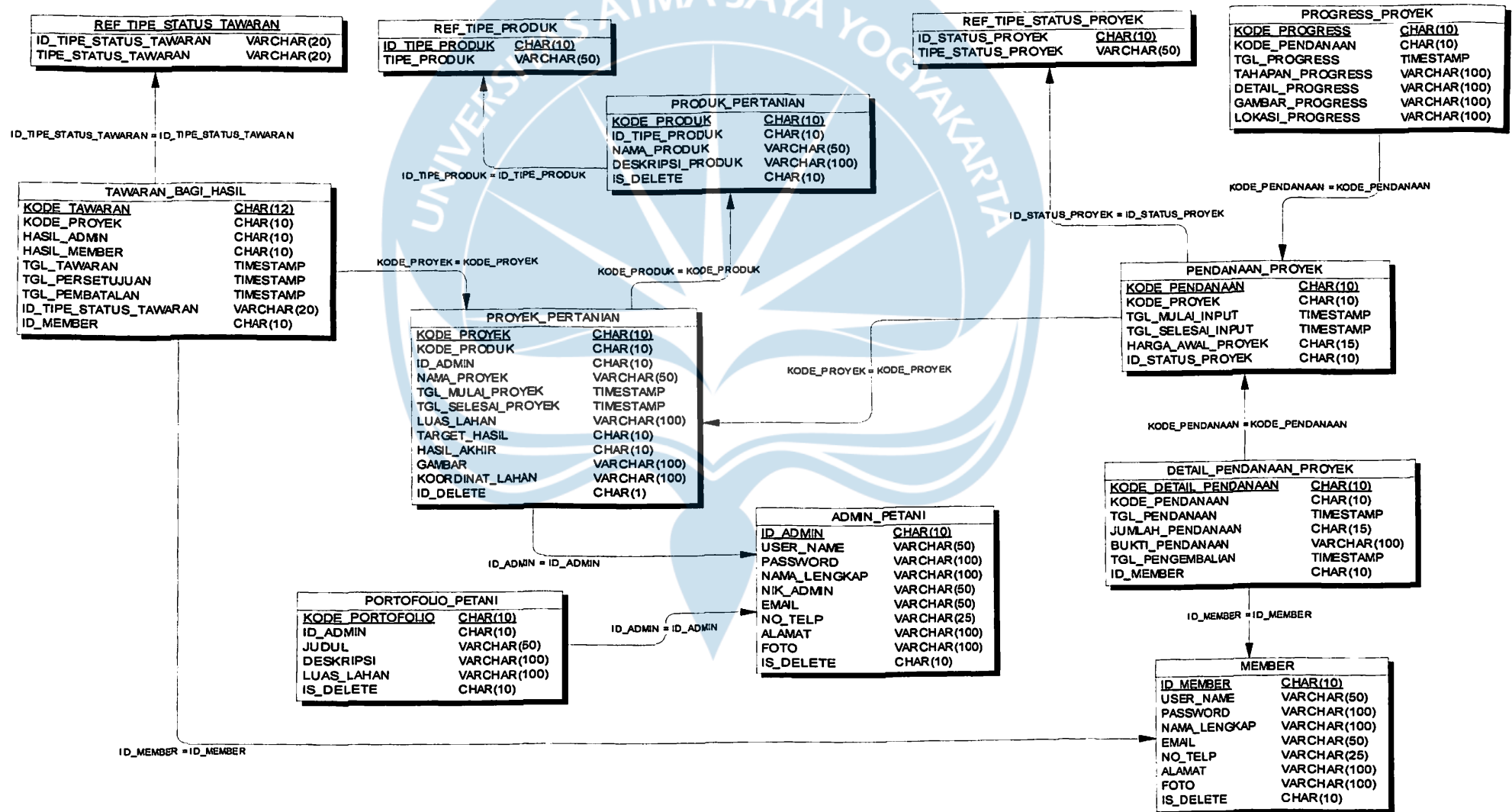
Tabel 11 : Struktur Tabel portofolio\_petani

Nama Field	Tipe Data	Null	Konstrain	Range Nilai	Default	Keterangan
Kode_portofolio	Varchar (25)	NO	Generate sistem	25		PRIMARY KEY
Id_admin	Char (10)	NO		10		FOREIGN KEY dari tabel admin_petani
Judul	Varchar (50)	YES		1-50		
Deskripsi	Varchar (100)	YES		1 – 100		
Luas_lahan	Varchar (100)	YES		1 – 100		
Hasil_panen	double	NO	Format angka 0 - 9			
Is_delete	Char (10)	YES		1 - 10	0	

Tabel 12 : Struktur Tabel

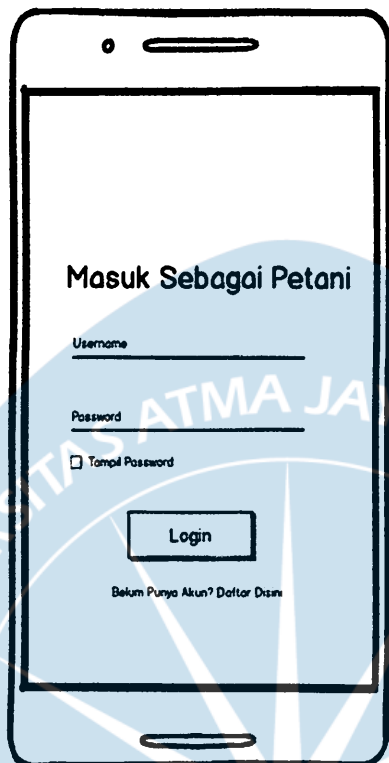
Nama Field	Tipe Data	Null	Konstrain	Range Nilai	Default	Keterangan
Kode_tawaran	Varchar (25)	NO	Generate sistem	25		PRIMARY KEY
Kode_proyek	Varchar (25)	NO		25		FOREIGN KEY dari tabel proyek_pertanian
Id_member	Char (15)	YES		15		FOREIGN KEY dari tabel member
Hasil_admin	Char (10)	NO	Format angka 1 – 100	1 – 10		
Hasil_member	Char (10)	NO	Format angka 1 – 100	1 – 10		
Tgl_tawaran	date	NO	SYSDATE		0000-00-00	
Tgl_persetujuan	date	YES	SYSDATE tidak boleh lebih dari tgl_tawaran		0000-00-00	
Tgl_pembatalan	date	YES	SYSDATE tidak boleh lebih dari tgl_tawaran		0000-00-00	
status	Int (10)	NO		1 -10	1	FOREIGN KEY dari tabel ref_tipe_status_tawaran

### 2.3.2 Physical Data Model



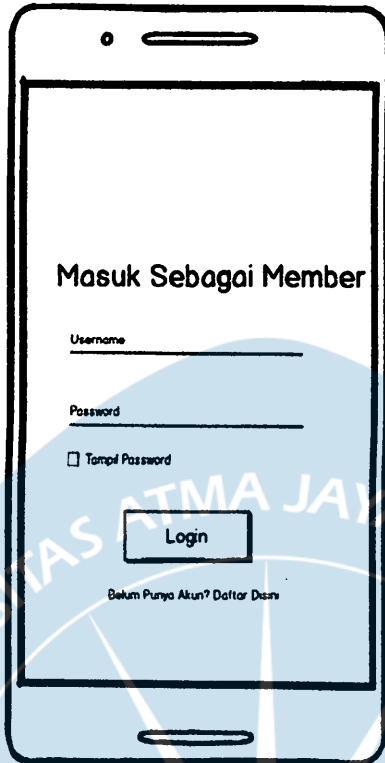
## 4 Perancangan Antar Muka

### Antarmuka Login Petani



- Deskripsi** : Antarmuka Login Pengguna memungkinkan Petani untuk masuk ke aplikasi. Antarmuka ini berisi beberapa data input dan tombol eksekusi yang terkait dengan layanan login petani.
- Input** :
  - Username dan Password tidak boleh kosong.
  - Minimal 6 karakter
- Output** : Masuk ke aplikasi sebagai petani
- Method/Algoritma** : **OnClick Button 'Login' / btnLogin**
  1. Membaca data username dan password.
  2. Melakukan validasi data input.
  3. Jika ada yang tidak valid, menampilkan peringatan dalam bentuk toast 'Username / password salah', kembali ke langkah 1.
  4. Mengkonversi password ke format SHA-1.
  5. Memanggil prosedur LoginAdmin() untuk meminta request API.
  6. API mendapatkan data petani (id admin, username, password, nama lengkap, no identitas, email, no telepon, alamat, foto) dari tabel admin\_petani.
  7. Mendapatkan response API berupa data petani.
  8. Masuk ke halaman utama petani.**Click Textview 'Belum Punya Akun'**
  1. Tampilan bergeser ke fragment Registrasi Petani

## 2. Antarmuka Login Member



- Deskripsi** : Antarmuka Login Pengguna memungkinkan Member untuk masuk ke aplikasi. Antarmuka ini berisi beberapa data input dan tombol eksekusi yang terkait dengan layanan login member.
- Input** :
  - Username dan Password tidak boleh kosong.
  - Minimal 6 karakter
- Output** : Masuk ke aplikasi sebagai member
- Method/Algoritma** : **Click Button 'Login' / btnLogin**
  1. Membaca data username dan password.
  2. Melakukan validasi data input.
  3. Jika ada yang tidak valid, menampilkan peringatan dalam bentuk toast 'Username / password salah', kembali ke langkah 1.
  4. Mengkonversi password ke format SHA-1.
  5. Memanggil prosedur LoginMember() untuk meminta request API.
  6. API mendapatkan data petani (id member, username, password, nama lengkap, no identitas, email, no telepon, alamat, foto) dari tabel member.
  7. Mendapatkan response API berupa data petani.
  8. Masuk ke halaman utama member.**Click Textview 'Belum Punya Akun'**
  1. Tampilan bergeser ke fragment Registrasi Petani



### 3. Antarmuka Registrasi Petani

The image shows a mobile application registration screen. At the top, the title 'Daftar' is centered. Below it are six input fields, each with a label and a horizontal line for text entry: 'Nama Lengkap', 'No Identitas', 'Email', 'Username', 'Password', and 'Konfirmasi Password'. Below the input fields is a rectangular button labeled 'Daftar'. At the bottom of the screen, there is a text link that says 'Sudah Punya Akun? Login Disini'. The entire screen is overlaid on a large, faint watermark of the Universitas Atma Jaya Yogyakarta logo.

- Deskripsi** : Antarmuka Registrasi Pengguna memungkinkan Petani untuk mendaftar ke aplikasi. Antarmuka ini berisi beberapa data input dan tombol eksekusi yang terkait dengan layanan register petani.
- Input** :
- Nama lengkap, Email, NIK, Username, Password tidak boleh kosong
  - Format email harus benar
  - NIK berupa angka 16 digit
  - Username dan password minimal 6 karakter
- Output** : Data tersimpan didatabase
- Method/Algoritma** : **Click Button 'Daftar' / btnDaftar**
1. Membaca data input.
  2. Melakukan validasi data input.
  3. Jika ada yang tidak valid, menampilkan peringatan dalam bentuk toast 'Data yang dimasukkan salah', kembali ke langkah 1.
  4. Memanggil prosedur RegisterPetani() untuk mengirimkan request.
  5. Api mengubah format password menjadi SHA-1.
  6. Api menggenerate id\_admin dan memasukkan id\_admin, user\_name, password, nama\_lengkap, dan email ke dalam tabel admin\_petani.
  7. Mendapatkan response API berupa status dan status message.
  8. Jika status = 0, maka data belum tersimpan. Jika status = 1, maka data telah tersimpan.
  9. Masuk ke halaman login petani.
- Click Textview 'Sudah Punya Akun'**
1. Tampilan bergeser ke fragment Login Petani

## Antarmuka Registrasi Member

The image shows a mobile application interface for member registration. The screen is titled "Daftar" (Register). It contains five input fields: "Nama Lengkap" (Full Name), "Email", "Username", "Password", and "Konfirmasi Password" (Confirm Password). Below the fields is a "Daftar" button. At the bottom, there is a link that says "Sudah Punya Akun? Login Di sini" (Already have an account? Login here).

**ripsi** : Antarmuka Registrasi Pengguna memungkinkan Member untuk masuk ke aplikasi. Antarmuka ini berisi beberapa data input dan tombol eksekusi yang terkait dengan layanan register member.

- t** :
- Nama lengkap, Email, NIK, Username, Password tidak boleh kosong
  - Format email harus benar
  - NIK berupa angka 16 digit
  - Username dan password minimal 6 karakter

**out** : Data tersimpan didatabase

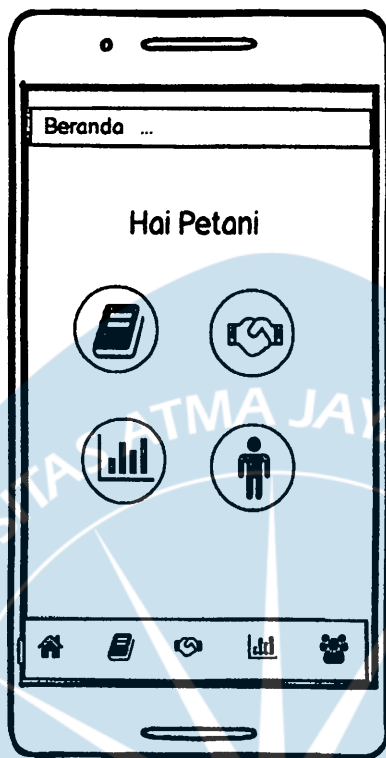
**hod/Algoritma** : **Click Button 'Daftar' / btnDaftar**

1. Membaca data input.
2. Melakukan validasi data input.
3. Jika ada yang tidak valid, menampilkan peringatan dalam bentuk toast 'Data yang dimasukkan salah', kembali ke langkah 1.
4. Memanggil prosedur RegisterMember() untuk mengirimkan request.
5. Api mengubah format password menjadi SHA-1.
6. Api menggenerate id\_member dan memasukkan id\_member, user\_name, password, nama\_lengkap, dan email ke dalam tabel member.
7. Mendapatkan response API berupa status dan status message.
8. Jika status = 0, maka data belum tersimpan. Jika status = 1, maka data telah tersimpan.
9. Masuk ke halaman login member.

**Click Textview 'Sudah Punya Akun'**

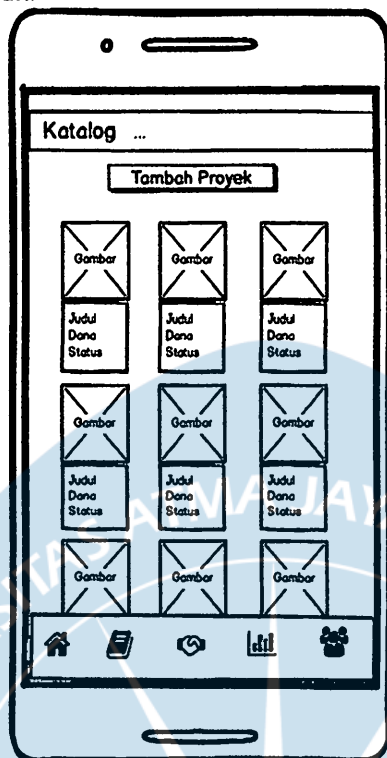
1. Tampilan bergeser ke fragment Login Member

## 5. Antarmuka Halaman Utama Petani dan Member



Deskripsi	: Antarmuka halaman utama digunakan untuk menampilkan fungsi-fungsi yang tersedia pada aplikasi. Antarmuka ini berisi beberapa button pada layar tengah dan bottom navbar yang berisi fungsi-fungsi utama.
Input	: -
Output	: Menampilkan button utama
Method/Algoritma	: <b>Click Button 'Gambar Buku' / btnKatalog</b> 1. Tampilan Fragment home berubah ke fragment katalog_admin <b>Click Button 'Gambar Salaman' / btnTawaran</b> 1. Tampilan Fragment home berubah ke fragment tawaran <b>Click Button 'Gambar Statistik' / btnProses</b> 1. Tampilan Fragment home berubah ke fragment proses <b>Click Button 'Gambar Orang' / btnProfil</b> 1. Tampilan Fragment home berubah ke fragment profil

## 6. Antarmuka Halaman Katalog Petani



### Deskripsi

: Antarmuka halaman katalog petani digunakan untuk menampilkan proyek-proyek yang telah dibuat oleh petani tersebut. Antarmuka ini berisi button “Tambah Proyek” dan cardview yang dibagi menjadi 3 kolom. Pada cardview menampilkan gambar dari proyek yang dibuat, judul, jumlah dana yang dibutuhkan, dan status dari proyek tersebut.

### Input

: -

### Output

: Menampilkan halaman katalog

### Method/Algoritma

: onCreate()

1. Aplikasi menyimpan data petani ke dalam shared preference pada saat login.
2. Kelas KatalogAdminFragment memanggil shared preference untuk mendapatkan id\_admin yang sedang login.
3. Memanggil prosedur showProject() untuk melakukan request ke api.
4. Api mengambil data – data dari banyak tabel sekaligus berdasarkan id admin yang dikirim request. Api mengambil kode pendanaan, kode proyek, kode produk, id admin, nama proyek, tgl mulai proyek, tanggal selesai proyek, luas lahan, target hasil, gambar, dan koordinat lahan dari tabel proyek pertanian. Ditambah data harga awal proyek dan status proyek dari tabel pendanaan proyek, ditambah data nama produk, deskripsi produk dan tipe produk dari tabel produk pertanian.
5. Menerima reponse berupa list proyek petani berdasarkan id\_admin yang dikirimkan saat melakukan request.

6. Response dikirim ke kelas RecyclerViewAdmin untuk dapat menampilkan hasil tersebut kedalam cardview.

7. Dalam kelas RecyclerViewAdmin list response ditampilkan pada cardview berdasarkan position/array list tersebut.

8. List response sudah ditampilkan ke halaman katalog.

**Click Cardview Tertentu / Cardview.Onclick**

1. Pada kelas RecyclerViewAdmin data proyek pada cardview yang diklik, disimpan ke shared preference PREF\_EDIT\_PROYEK.

2. Tampilan berpindah ke kelas Detail Katalog .

**Click Button 'Tambah Proyek' / btnTambah**

1. Tampilan berpindah ke kelas InputProyek Activity.

## 7. Antarmuka Detail Proyek (Petani)



**Deskripsi** : Antarmuka halaman detail katalog petani digunakan untuk menampilkan detail proyek yang dipilih dari halaman sebelumnya. Antarmuka ini berisi button "Ubah Proyek" dan "Hapus". Juga berisi detail lengkap dari proyek yang dipilih.

**Input** : -

**Output** : Menampilkan halaman detail katalog

**Method/Algoritma** : onCreate()

1. Memanggil shared preference PREF\_EDIT\_PROYEK untuk mengambil semua data yang disimpan pada preference tersebut.

2. Data tersebut ditampilkan pada masing-masing textview dan imageview yang sudah dibuat.

3. Memanggil prosedur `getTawaranHasil` untuk mengambil tawaran yang mana yang masih berlangsung pada proyek tersebut.

#### **onClick Button Edit Proyek**

1. Tampilan berpindah ke kelas `EditProyekActivity`

#### **onClick Button Hapus**

1. Menampilkan alert dialog. Jika pengguna memilih yes maka lanjut ke langkah 2.

2. Memanggil prosedur hapus proyek.

3. Mendapatkan `kodeProyek` yang disimpan pada preference.

4. Memanggil method `deleteproyek` untuk request api.

5. Method `delete` tersebut mengubah nilai `is_delete` dari 0 menjadi 1 pada tabel `proyek_pertanian` untuk `kodeProyek` yang dikirimkan request.

6. Api mengirimkan response berupa status dan status message.

7. Aplikasi membaca response, jika status response 1 maka menampilkan toast status message "berhasil dihapus".

8. Kemudian kembali ke halaman katalog petani.

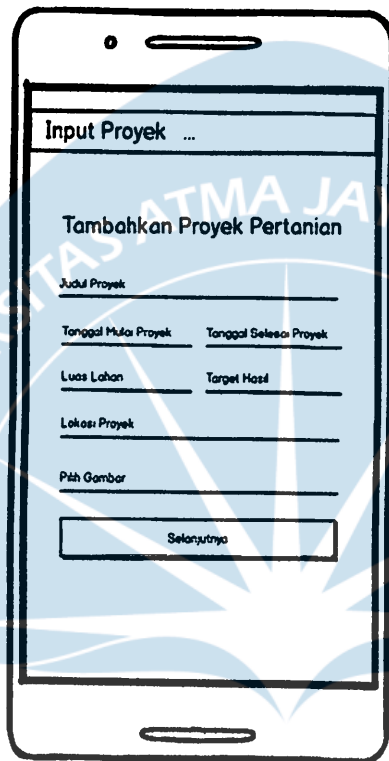
Antarmuka Input Bagi Hasil



Deskripsi : Antarmuka halaman input proyek bagian pertama. Digunakan untuk menginputkan proyek baru. Antarmuka ini berisi beberapa data input dan button selanjutnya.

Output : Nama produk, jenis produk, dan deskripsi produk

- Output : -
- Method/Algoritma : **onClick() Button Selanjutnya**
1. Mengecek validasi inputan.
  2. Jika ada input yang salah, menampilkan toast "Masukkan ada yang salah".
  3. Jika sudah benar maka data input disimpan ke shared preference PREF\_INPUT\_PROYEK
  4. Tampilan berpindah ke kelas InputProject2.



- Deskripsi : Antarmuka halaman input proyek bagian kedua. Digunakan untuk menginputkan proyek baru. Antarmuka ini berisi beberapa data input dan button selanjutnya.
- Input : Judul Proyek, Tanggal mulai proyek, Tanggal selesai proyek, luas lahan, target hasil, lokasi proyek, dan gambar.
- Output : -
- Method/Algoritma : **onClick() Button Selanjutnya**
1. Mengecek validasi inputan.
  2. Jika ada input yang salah, menampilkan toast "Masukkan ada yang salah".
  3. Jika sudah benar maka data input disimpan ke shared preference PREF\_INPUT\_PROYEK
  4. Tampilan berpindah ke kelas InputProject3.



- Deskripsi** : Antarmuka halaman input proyek bagian ketiga. Digunakan untuk menginputkan proyek baru. Antarmuka ini berisi beberapa data input dan button konfirmasi proyek.
- Input** : Jumlah dana yang dibutuhkan.
- Output** : -
- Method/Algoritma** : **onClick() Button Konfirmasi**
1. Mengecek validasi inputan.
  2. Jika ada input yang salah, menampilkan toast "Masukkan ada yang salah".
  3. Jika input sudah benar maka semua data input yang telah disimpan pada preference di panggil dengan menggunakan prosedur `getData()`.
  4. Prosedur `getData()` mengambil semua data yang tersimpan di shared preference dan kemudian dimasukkan ke variable yang telah di deklarasikan.
  5. Memanggil prosedur `generateKodeProyek()`.
  6. Prosedur `generateKodeProyek()` mengirimkan request ke api dengan method `getKodeProyek`. Kemudian didalam api request ditangani oleh fungsi `generate_kode_proyek` yang kemudian dikirimkan response berupa kode proyek yang telah digenerate.
  7. Kode proyek disimpan ke preference.
  8. Aplikasi menerima response api dan kemudian dilanjutkan dengan memanggil prosedur `inputproject()` untuk memasukkan data ke database.
  9. Memanggil prosedur `inputproject()`. Prosedur `inputproject()` digunakan untuk memasukkan data ke dalam database melalui api.
  10. Prosedur mengirim request post proyek. Data yang dikirimkan berupa `nama_produk`, `deskripsi_produk`, `id_tipe_produk`, `id_admin`, `kode_proyek`,



nama\_proyek, luas\_lahan, target\_hasil, lokasi\_proyek, gambar, tgl\_mulai\_proyek, tgl\_selesai\_proyek, harga\_proyek.

11. Didalam api, request tersebut ditangani oleh beberapa fungsi sekaligus.

12. Fungsi pertama adalah fungsi post\_produk(). Fungsi ini untuk menyimpan data produk ke database.

13. Pada fungsi post\_produk(), kode\_produk digenerate dengan fungsi generateKodeProduk(), lalu data-data yang berkaitan produk (kode\_produk, nama\_produk, id\_tipe\_produk, dan deskripsi\_produk) dimasukkan ke tabel produk\_pertanian.

14. Jika data produk sudah tersimpan, selanjutnya memanggil fungsi post\_proyek().

15. Pada fungsi post\_proyek(), kode proyek yang telah digenerate dan data-data terkait proyek (kode\_proyek, nama\_proyek, tgl\_mulai\_proyek, tgl\_selesai\_proyek, luas\_lahan, target hasil, gambar, dan is\_delete yang di default nilainya 1) dimasukkan ke tabel proyek\_pertanian.

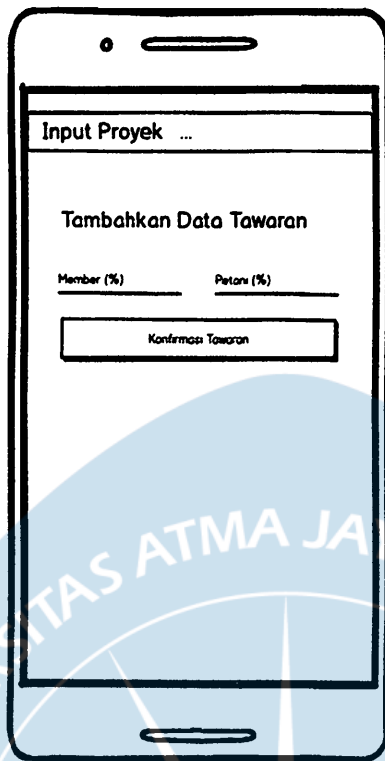
16. Jika data proyek sudah tersimpan, selanjutnya memanggil fungsi post\_pendanaan().

17. Pada fungsi post\_pendanaan(), kode\_pendanaan digenerate dengan fungsi generateKodePendanaan(), kemudian tanggal input pendanaan yang dimasukkan adalah tanggal memasukkan data ke database, lalu jumlah dana yang dibutuhkan diambil dari data yang dikirim lewat request, dan status proyek diberi nilai default yaitu 1, sehingga status proyek nantinya adalah belum didanai. Kemudian data-data tersebut disimpan ke tabel pendanaan\_proyek.

18. Jika semua data sudah disimpan maka api mengirimkan response berupa status dan status message.

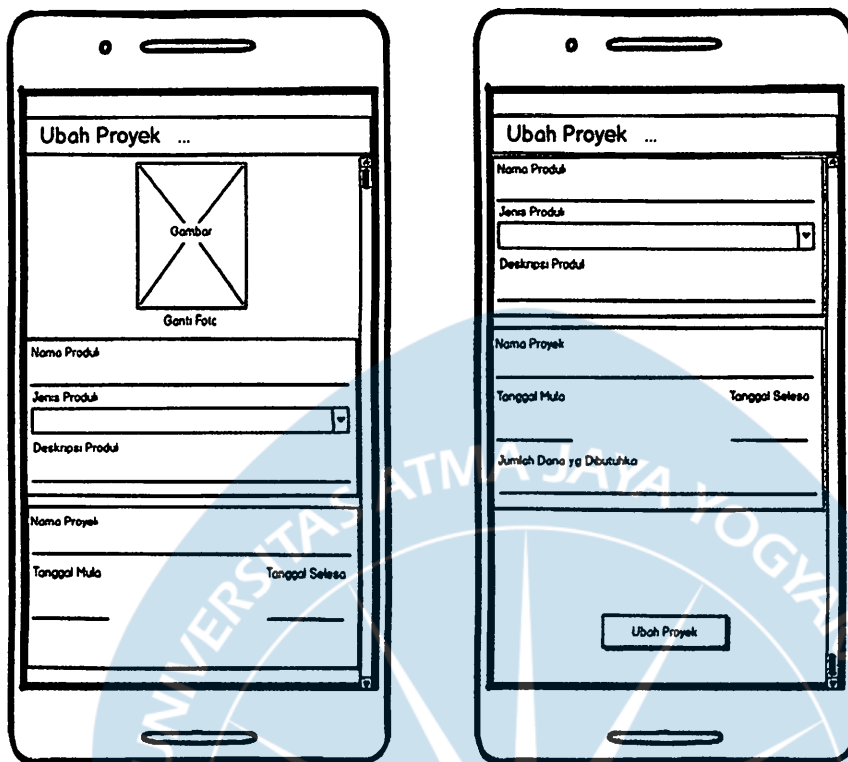
19. Aplikasi membaca response, jika status response 1 maka menampilkan toast status message "berhasil ditambahkan".

20. Tampilan berpindah ke kelas InputProject4.



- Deskripsi** : Antarmuka halaman input proyek bagian keempat. Digunakan untuk menginputkan proyek baru. Antarmuka ini berisi beberapa data input dan button konfirmasi tawaran.
- Input** : Persen Member dan Persen Petani.
- Output** : -
- Method/Algoritma** : **onClick() Button Konfirmasi**
1. Mengecek validasi inputan.
  2. Jika ada input yang salah, menampilkan toast "Masukkan ada yang salah".
  3. Jika jumlah input member ditambah input petani tidak sama dengan 100, menampilkan toast "Jumlah belum 100".
  4. Jika data sudah benar memanggil prosedur `inputTawaran()`.
  5. Prosedur `inputTawaran()` melakukan request ke api. Data yang dikirim yaitu `kode_proyek` yang sudah digenerate, persen hasil petani, dan persen hasil member.
  6. Didalam api, request ditangani oleh fungsi `post_tawaran`. `Kode_tawaran` digenerate dengan fungsi `generateKodeTawaran()`, tanggal tawaran didapat dari tanggal tawaran dimasukkan, hasil admin dan hasil member didapat dari data yang dikirim lewat request, kemudian data – data tersebut dimasukkan ke tabel tawaran bagi hasil.
  7. Jika semua data sudah disimpan maka api mengirimkan response berupa status dan status message.
  8. Aplikasi membaca response, jika status response 1 maka menampilkan toast status message "berhasil ditambahkan".
  9. Tampilan berpindah ke kelas `KatalogAdminFragment`.

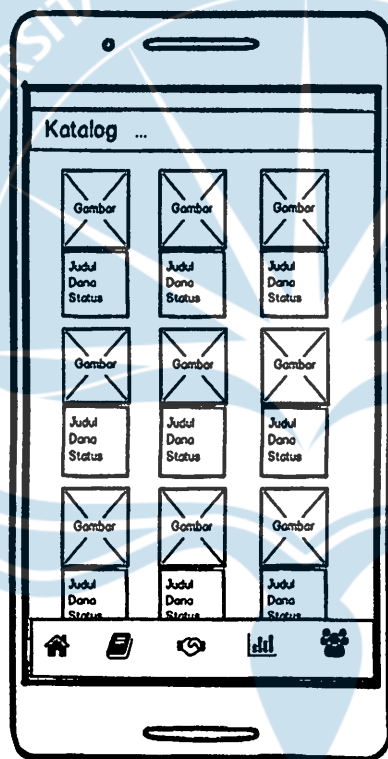
## Antarmuka Halaman Edit Proyek



- Deskripsi** : Antarmuka halaman edit proyek. Digunakan untuk mengubah data proyek lama. Antarmuka ini berisi beberapa data input dan button ubah proyek.
- Input** : Nama produk, jenis produk, deskripsi produk, nama proyek, tanggal mulai, tanggal selesai, dan jumlah dana yang dibutuhkan.
- Output** : Data terupdate didatabase
- Method/Algoritma** : **onCreate()**
1. Menampilkan data yang sudah disimpan di preference ke layout dengan menggunakan prosedur `setLayout()`.
- onClick() Button Ubah**
1. Mengecek validasi inputan.
  2. Jika ada input yang salah, menampilkan toast "Masukkan ada yang salah".
  4. Jika data sudah benar memanggil prosedur `updateProyek()`.
  5. Prosedur `updateProyek()` melakukan request ke api. Data yang dikirim yaitu kode produk, kode proyek, kode pendanaan, dan data data yang di akan diubah.
  6. Didalam api, request ditangani oleh beberapa fungsi sekaligus. Fungsi pertama adalah update produk.
  7. Fungsi `updateProduk` mengupdate data nama produk, id tipe produk, dan deskripsi produk berdasarkan kode produk yang dikirim request.
  8. Memanggil fungsi `update_proyek()`.

9. Fungsi update proyek mengupdate data nama proyek, tanggal mulai proyek, tanggal selesai proyek, target hasil, luas lahan dan gambar berdasarkan dengan kode proyek yang dikirim request.
10. Memanggil fungsi update pendanaan.
11. Fungsi update pendanaan mengupdate harga awal proyek berdasarkan kode pendanaan yang dikirim request.
12. Jika semua data sudah disimpan maka api mengirimkan response berupa status dan status message.
13. Aplikasi membaca response, jika status response 1 maka menampilkan toast status message “berhasil diubah”.
14. Tampilan berpindah ke kelas DetailKatalogAdmin.

Antarmuka Halaman Katalog Member



- deskripsi : Antarmuka halaman Katalog member. Digunakan untuk menampilkan proyek-proyek yang terdapat dalam database.
- Output : -
- Input : Proyek-proyek dalam database
- Method/Algoritma : onCreate()
1. Aplikasi menyimpan data member ke dalam shared preference pada saat login.
  2. Memanggil shared preference untuk mendapatkan id\_member yang sedang login.
  3. Memanggil prosedur showProject() untuk melakukan request ke api.

4. Api mengambil data – data dari banyak tabel sekaligus. Api mengambil kode pendanaan, kode proyek, kode produk, id admin, nama proyek, tgl mulai proyek, tanggal selesai proyek, luas lahan, target hasil, gambar, dan koordinat lahan dari tabel proyek pertanian. Ditambah data harga awal proyek dan status proyek dari tabel pendanaan proyek, ditambah data nama produk, deskripsi produk dan tipe produk dari tabel produk pertanian.

5. Menerima reponse berupa list proyek petani.

6. Response dikirim ke kelas RecyclerViewMember untuk dapat menampilkan hasil tersebut kedalam cardview.

7. Dalam kelas RecyclerViewMember list response ditampilkan pada cardview berdasarkan position/array list tersebut.

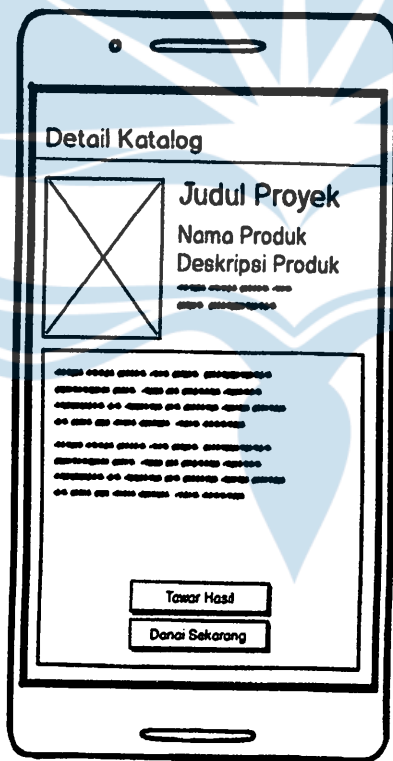
8. List response sudah ditampilkan ke halaman katalog.

#### Click Cardview Tertentu / Cardview.Onclick

1. Pada kelas RecyclerViewMember data proyek pada cardview yang diklik, disimpan ke shared preference PREF\_TRANSACTION.

2. Tampilan berpindah ke kelas Detail Katalog .

### 11. Antarmuka Halaman Detail Katalog Member



Deskripsi : Antarmuka halaman Detail Katalog member. Digunakan untuk menampilkan detail proyek yang dipilih dari halaman katalog. Antarmuka berisi button tawar hasil dan button danai sekarang.

Input : -

Output : -

/Algoritma : **onCreate()**

1. Memanggil shared preference PREF\_TRANSACTION untuk mengambil semua data yang disimpan pada preference tersebut.
2. Data tersebut ditampilkan pada masing-masing textview dan imageview yang sudah dibuat.
3. Memanggil prosedur getTawaranHasil untuk mengambil tawaran yang mana yang masih berlangsung pada proyek tersebut.

**onClick Button Tawar**

1. Tampilan berpindah ke kelas TawaranMemberActivity

**onClick Button Danai**

1. Tampilan berpindah ke kelas TransaksiActivity

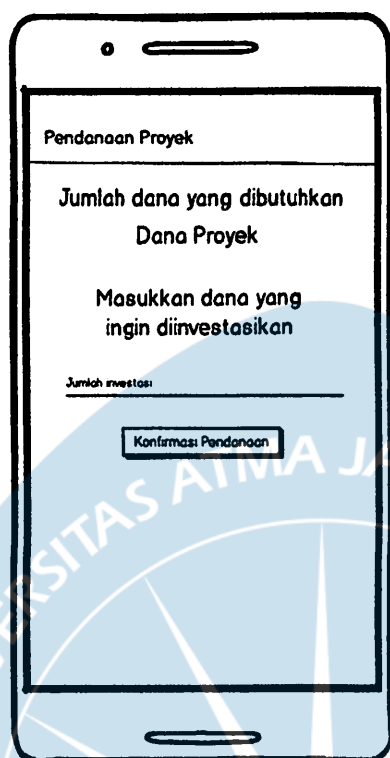
**onClick TextView ProfilAdmin**

1. Memanggil prosedur getDetailAdmin().
2. getDetailAdmin mengirim request dengan data yang dikirim adalah id admin pada proyek tersebut.
3. Api menerima request dan ditangani oleh fungsi getAdmin.
4. Fungsi getAdmin mengambil data petani seperti id admin, user name, password, nama lengkap, no identitas, email, no telepon, alamat dan foto dari tabel admin petani.
5. Api mengirim response ke aplikasi.
6. Aplikasi membaca response tersebut dan menampilkan ke layout.

**onClick Textview PortofolioAdmin**

1. Memanggil prosedur showPortofolio().
2. showPortofolio mengirim request dengan data yang dikirim adalah id admin pada proyek tersebut.
3. Api menerima request dan ditangani oleh fungsi getPortofolio.
4. Fungsi getPortofolio mengambil data portofolio petani berdasarkan id admin yang dikirim request. Data portofolio meliputi judul portofolio, deskripsi portofolio, luas lahan dan hasil panen. Data tersebut diambil dari tabel portofolio petani oleh api.
5. Api mengirim response berupa list portofolio petani ke aplikasi.
6. Aplikasi membaca response tersebut dan mengirimnya ke kelas RecyclerViewPortofolio.
7. Dalam kelas RecyclerViewPortofolio list response ditampilkan pada cardview berdasarkan position/array list tersebut.
8. List response sudah ditampilkan ke halaman Detail Katalog.

Antarmuka Halaman Transaksi Pendanaan



Pendanaan Proyek

Jumlah dana yang dibutuhkan  
Dana Proyek

Masukkan dana yang  
ingin diinvestasikan

Jumlah investasi: \_\_\_\_\_

Konfirmasi Pendanaan

- ripsi : Antarmuka halaman Transaksi Pendanaan. Digunakan untuk memasukkan transaksi baru oleh member. Antarmuka berisi button konfirmasi pendanaan.
- t :
  - Jumlah dana yang dimasukkan tidak boleh kurang dari harga proyek tersebut
  - Inputan jumlah dana berupa angka desimal
- ut : Data transaksi disimpan ke database
- od/Algoritma : **onClick() Button Konfirmasi**
  1. Mengecek validasi inputan.
  2. Jika ada input yang salah, menampilkan toast "Masukkan ada yang salah".
  3. Jika input sudah benar maka muncul alert dialog menanyakan "Apakah yakin melakukan investasi?".
  4. Jika pengguna memilih ya, lanjut memanggil prosedur inputInvestasi().
  5. Prosedur inputInvestasi() mengirim request ke api dengan data yang dikirim yaitu kode\_pendanaan, id\_member, jumlah pendanaan, dan tanggal pengembalian. Tanggal pengembalian sama dengan tanggal selesai proyek tersebut.
  6. API menerima request dan ditangani oleh fungsi post\_transaksi().
  7. Fungsi post\_transaksi memasukkan data kode\_detail\_transaksi yang telah digenerate oleh fungsi generateKodeDetailTransaksi(), kemudian data kode pendanaan, id member, jumlah pendanaan, tanggal pengembalian,

dan tanggal pendanaan yang diset tanggal sekarang. Data-data tersebut dimasukkan ke tabel detail pendanaan oleh api.

8. Aplikasi menerima response api dan kemudian dilanjutkan dengan memanggil prosedur `updateStatus()` untuk mengubah status proyek.

9. Prosedur `updateStatus()` mengirim request dengan data yang dikirim adalah kode pendanaan dan status proyek 2.

10. API menerima request dan request ditangani oleh fungsi update status proyek.

11. Fungsi update status proyek mengubah status proyek berdasarkan kode pendanaan yang dikirim pada tabel pendanaan proyek dari status 1 menjadi status 2 yaitu menjadi "Telah didanai".

12. API mengirimkan response berupa status dan status message.

13. Jika status berhasil diubah maka lanjut memanggil prosedur update status tawaran untuk mengubah status tawaran agar tidak bisa ditawar lagi.

14. Prosedur `updateStatusTawaran()` mengirim request dengan data yang dikirim adalah kode tawaran dan status tawaran.

15. API menerima request dan request ditangani oleh fungsi update status tawaran.

16. Fungsi update status tawaran mengubah status tawaran pada tabel tawaran bagi hasil menjadi 4 yaitu "Sudah didanai" berdasarkan kode tawaran yang dikirim request.

17. API mengirimkan response berupa status dan status message.

18. Jika status berhasil diubah maka lanjut memanggil prosedur update id member untuk mengubah id member pada tawaran tersebut agar tahu siapa yang sudah mendanai.

19. Prosedur `updateIdMember()` mengirim request dengan data yang dikirim adalah kode tawaran dan id member.

20. API menerima request dan request ditangani oleh fungsi update id member.

21. Fungsi update id member mengubah id member pada tabel tawaran bagi hasil berdasarkan kode tawaran yang dikirim request.

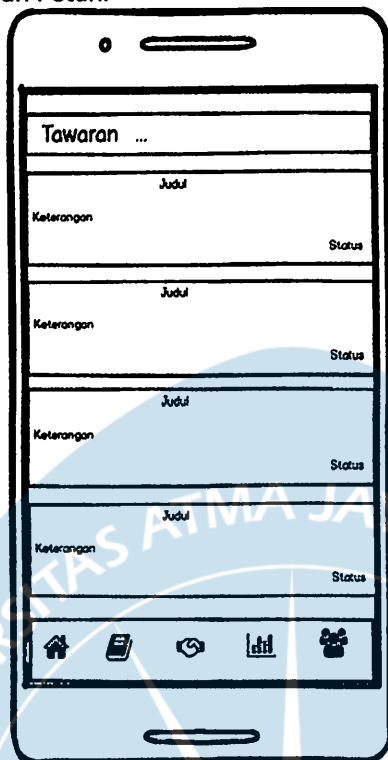
22. API mengirimkan response berupa status dan status message.

23. Jika id member berhasil ditambahkan, akan muncul toast yang berisi status message "Berhasil di danai".

24. Tampilan berpindah ke kelas `DetailTransaksi`.



### 3. Antarmuka Halaman List Tawaran Petani

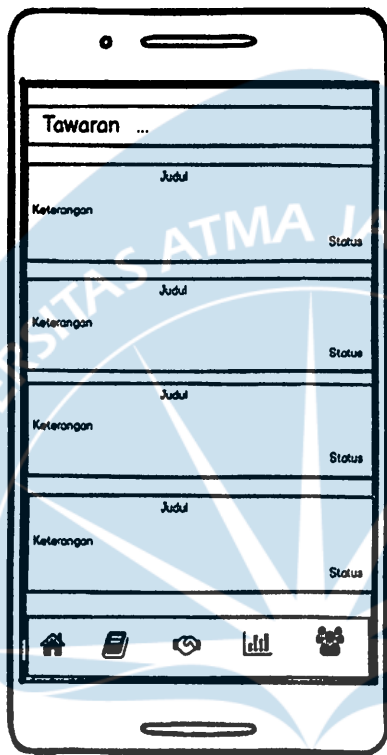


- Deskripsi** : Antarmuka halaman Tawaran List Petani. Digunakan untuk menampilkan tawaran dan statusnya untuk setiap proyek yang dimiliki petani.
- Input** : -
- Output** : Tawaran proyek petani tertentu.
- Method/Algoritma** : **onCreate()**
1. Memanggil shared preference untuk mendapatkan id\_admin yang sedang login.
  2. Memanggil prosedur showListTawaran() untuk melakukan request ke api.
  3. Api mengambil data – data berdasarkan id admin yang dikirim request dari banyak tabel sekaligus. Api mengambil data kode tawaran, kode proyek, id member, hasil admin, hasil member, tanggal tawaran, tanggal persetujuan, tanggal pembatalan dan status tawaran dari tabel tawaran bagi hasil yang digabung dengan data id admin dan nama proyek dari tabel proyek pertanian yang digabung lagi dengan tipe status tawaran dari tabel ref tipe status tawaran. Digabung lagi dengan data nama lengkap dari tabel member dan dari tabel pendanaan proyek nama proyek diambil data kode pendanaan.
  4. Response di kirim dalam bentuk list tawaran.
  5. Response dikirim ke kelas RecyclerViewTawaranAdmin untuk dapat menampilkan hasil tersebut kedalam listview.
  6. Dalam kelas RecyclerViewTawaranAdmin list response ditampilkan pada listview berdasarkan position/array list tersebut.
  7. List response sudah ditampilkan ke halaman tawaran petani.

### Click Listitem Tertentu / ListItem.Onclick

1. Pada kelas RecyclerViewTawaranAdmin data proyek pada listview yang diklik, disimpan ke shared preference PREF\_TAWARAN\_ADMIN.
2. Tampilan berpindah ke kelas TawaranAdminActivity.

### Antarmuka Halaman List Tawaran Member



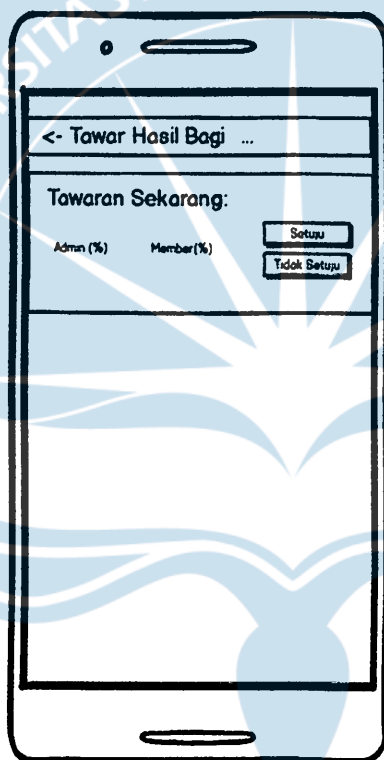
- Deskripsi** : Antarmuka halaman List Tawaran Member. Digunakan untuk menampilkan tawaran dan statusnya untuk setiap proyek yang dimiliki petani.
- Output** : -
- Input** : Tawaran proyek petani tertentu.
- Method/Algoritma** : **onCreate()**
1. Memanggil shared preference untuk mendapatkan id\_member yang sedang login.
  2. Memanggil prosedur showListTawaran() untuk melakukan request ke api.
  3. Api mengambil data – data berdasarkan id member yang dikirim request dari banyak tabel sekaligus. Api mengambil data kode tawaran, kode proyek, id member, hasil admin, hasil member, tanggal tawaran, tanggal persetujuan, tanggal pembatalan dan status tawaran dari tabel tawaran bagi hasil yang digabung dengan data id admin dan nama proyek dari tabel proyek pertanian yang digabung lagi dengan tipe status tawaran dari tabel ref tipe status tawaran. Digabung lagi dengan data nama lengkap dari tabel member dan dari tabel pendanaan proyek nama proyek diambil data kode pendanaan.

4. Response di kirim dalam bentuk list tawaran.
5. Response dikirim ke kelas RecyclerViewTawaranMember untuk dapat menampilkan hasil tersebut kedalam listview.
6. Dalam kelas RecyclerViewTawaranMember list response ditampilkan pada listview berdasarkan position/array list tersebut.
7. List response sudah ditampilkan ke halaman tawaran member.

#### Click ListItem Tertentu / ListItem.Onclick

1. Pada kelas RecyclerViewTawaranMember data proyek pada listview yang diklik, disimpan ke shared preference PREF\_TAWARAN\_MEMBER.
2. Tampilan berpindah ke kelas TawaranMemberActivity.

#### 5. Antarmuka Halaman Tawaran



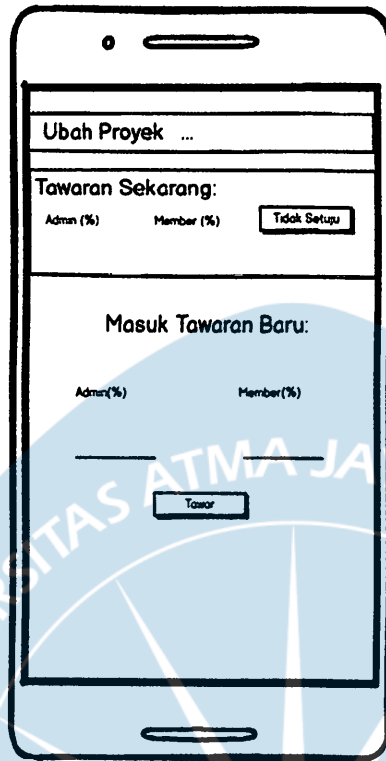
- Deskripsi** : Antarmuka halaman Tawaran. Digunakan untuk menampilkan tawaran yang statusnya "Menunggu". Antarmuka berisi button Setuju dan button Tidak Setuju.
- Input** : Persen Hasil Admin dan Persen Hasil Member
- Output** : -
- Method/Algoritma** : **onCreate()**
1. Memanggil shared preference untuk mendapatkan persen hasil petani dan persen hasil member.
  2. Menampilkannya ke textview.
- onClick() Button Setuju**

1. Menghilangkan button tidak setuju, dengan men-*set visibility* menjadi *gone*.
2. Memanggil prosedur `updateStatusProyek()`, untuk mengupdate status proyek yang disetujui.
3. Prosedur `updateStatusProyek` mengirim request api dengan data yang dikirim kode pendanaan dan status proyek.
4. API menerima request dan request ditangani oleh fungsi update status proyek.
5. Fungsi update status proyek mengubah status proyek berdasarkan kode pendanaan yang dikirim pada tabel pendanaan proyek dari status 1 menjadi status 5 yaitu menjadi "Sedang Ditawar".
6. API mengirimkan response berupa status dan status message.
7. Jika status berhasil diubah maka lanjut memanggil prosedur update status tawaran untuk mengubah status tawaran agar tidak bisa ditawar lagi.
8. Prosedur `updateStatusTawaran()` mengirim request dengan data yang dikirim adalah kode tawaran dan status tawaran.
9. API menerima request dan request ditangani oleh fungsi update status tawaran.
10. Fungsi update status tawaran mengubah status tawaran pada tabel tawaran bagi hasil menjadi 2 yaitu "Setuju" berdasarkan kode tawaran yang dikirim request.
11. API mengirimkan response berupa status dan status message.
12. Jika status berhasil diubah maka akan menampilkan text view jika tawaran sudah di setujui.

#### **onClick() Button Tidak Setuju**

1. Menghilangkan button setuju, dengan men-*set visibility* menjadi *gone*.
2. Memanggil fragment Tawaran Tidak Setuju.
3. Tampilan berpindah ke fragment Tawaran Tidak Setuju.

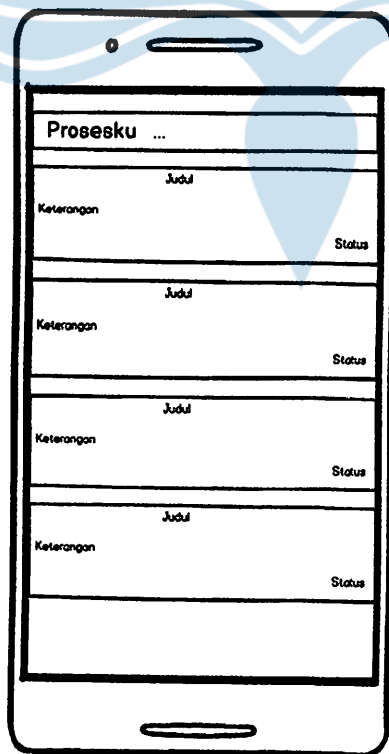
## 16. Antarmuka Halaman Tawaran yang Tidak di Setujui



- Deskripsi** : Antarmuka halaman Tawaran. Digunakan untuk memasukkan tawaran baru baik untuk member maupun petani.
- Input** : Persen Hasil Admin dan Persen Hasil Member
- Output** : Tawaran baru disimpan
- Method/Algoritma** : **onClick()** Button Tawar
1. Mengecek validasi inputan.
  2. Jika ada input yang salah maka muncul toast "Inputan salah". Jika jumlah admin dan member tidak sama dengan 100 maka akan muncul toast "Jumlah harus 100". Lalu kembali ke langkah 1.
  3. Memanggil prosedur `updateStatusProyek()`, untuk mengupdate status proyek yang disetujui.
  4. Prosedur `updateStatusProyek` mengirim request api dengan data yang dikirim kode pendanaan dan status proyek.
  5. API menerima request dan request ditangani oleh fungsi update status proyek.
  6. Fungsi update status proyek mengubah status proyek berdasarkan kode pendanaan yang dikirim pada tabel pendanaan proyek dari status 1 menjadi status 5 yaitu menjadi "Sedang Ditawar".
  7. API mengirimkan response berupa status dan status message.
  8. Jika status berhasil diubah maka lanjut memanggil prosedur update status tawaran untuk mengubah status tawaran agar tidak bisa ditawar lagi.

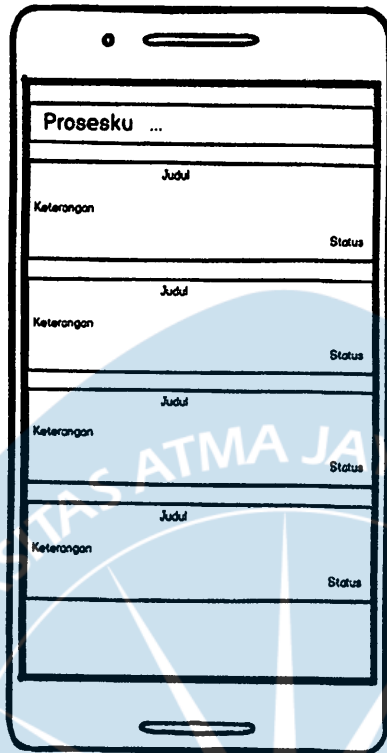
9. Prosedur `updateStatusTawaran()` mengirim request dengan data yang dikirim adalah kode tawaran dan status tawaran.
10. API menerima request dan request ditangani oleh fungsi update status tawaran.
11. Fungsi update status tawaran mengubah status tawaran pada tabel tawaran bagi hasil menjadi 3 yaitu “ Tidak Setuju” berdasarkan kode tawaran yang dikirim request.
12. API mengirimkan response berupa status dan status message.
13. Jika status berhasil lanjut memanggil prosedur `inputTawaranBaru()` untuk menambahkan tawaran baru.
14. Prosedur `inputTawaranBaru()` mengirim request dengan data yang dikirim adalah kode proyek, id member, hasil admin, dan hasil member.
15. Didalam api, request ditangani oleh fungsi `post_tawaran`. Kode\_tawaran digenerate dengan fungsi `generateKodeTawaran()`, tanggal tawaran didapat dari tanggal tawaran dimasukkan, hasil admin dan hasil member didapat dari data yang dikirim lewat request, kemudian data – data tersebut dimasukkan ke tabel tawaran bagi hasil.
7. Jika semua data sudah disimpan maka api mengirimkan response berupa status dan status message.
8. Aplikasi membaca response, jika status response 1 maka menampilkan toast status message “berhasil ditambahkan”.
9. Tampilan berpindah ke kelas `ListTawaran`.

## 17. Antarmuka Halaman List Proyek Petani



- Deskripsi** : Antarmuka halaman List Proyek Petani. Digunakan untuk menampilkan proyek – proyek petani yang telah didanai maupun sedang menunggu konfirmasi petani.
- Input** : -
- Output** : List proyek yang sudah di proses.
- Method/Algoritma** : **onCreate()**
1. Memanggil shared preference untuk mendapatkan id\_admin yang sedang login.
  2. Memanggil prosedur showListProject() untuk melakukan request ke api.
  3. Api mengambil data – data berdasarkan id admin yang dikirim request dari banyak tabel sekaligus. Api mengambil data kode proyek, kode produk, id admin, nama proyek, tanggal mulai proyek, tanggal selesai proyek, dan gambar dari tabel proyek pertanian. Kemudian digabungkan dengan data kode pendanaan, harga awal proyek, status proyek dari tabel pendanaan proyek. Digabungkan lagi dengan id member, jumlah pendanaan, kode detail transaksi dari tabel detail pendanaan.
  4. Response di kirim dalam bentuk list proyek.
  5. Response dikirim ke kelas Recycler List View Proyek Admin untuk dapat menampilkan hasil tersebut kedalam listview.
  6. Dalam kelas Recycler List View Proyek Admin list response ditampilkan pada listview berdasarkan position/array list tersebut.
  7. List response sudah ditampilkan ke halaman List Proyek Admin.
- Click ListItem Tertentu / ListItem.Onclick**
1. Pada kelas Recycler List View Proyek Admin data proyek pada listview yang diklik, disimpan ke shared preference PREF\_PROGRESS.
  2. Tampilan berpindah ke kelas Progresses Admin Activity.

## 18. Antarmuka Halaman List Proyek Member



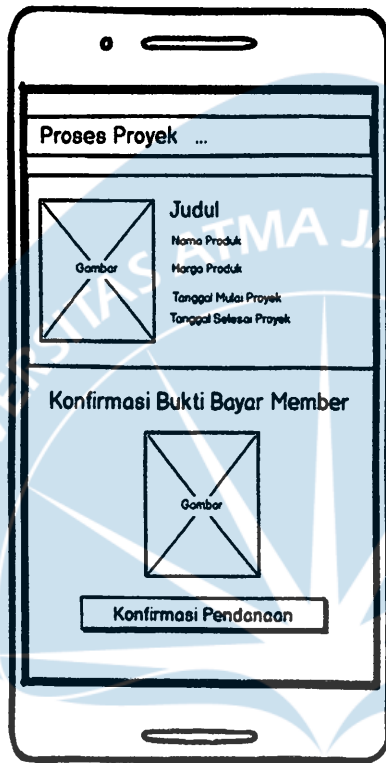
- Deskripsi** : Antarmuka halaman List Proyek Member. Digunakan untuk menampilkan proyek – proyek member yang telah didanai maupun sedang menunggu konfirmasi member.
- Input** : -
- Output** : List proyek yang sudah di proses.
- Method/Algoritma** : **onCreate()**
1. Memanggil shared preference untuk mendapatkan id\_member yang sedang login.
  2. Memanggil prosedur showListProject() untuk melakukan request ke api.
  3. Api mengambil data – data berdasarkan id member yang dikirim request dari banyak tabel sekaligus. Api mengambil data kode proyek, kode produk, id admin, nama proyek, tanggal mulai proyek, tanggal selesai proyek, dan gambar dari tabel proyek pertanian. Kemudian digabungkan dengan data kode pendanaan, harga awal proyek, status proyek dari tabel pendanaan proyek. Digabungkan lagi dengan id member, jumlah pendanaan, kode detail transaksi dari tabel detail pendanaan.
  4. Response di kirim dalam bentuk list proyek.
  5. Response dikirim ke kelas Recycler List View Proyek Member untuk dapat menampilkan hasil tersebut kedalam listview.
  6. Dalam kelas Recycler List View Proyek Member list response ditampilkan pada listview berdasarkan position/array list tersebut.
  7. List response sudah ditampilkan ke halaman List Proyek Member.



### Click ListItem Tertentu / ListItem.Onclick

1. Pada kelas Recycler List View Proyek Member data proyek pada listview yang diklik, disimpan ke shared preference PREF\_PROGRESS.
2. Tampilan berpindah ke kelas Progresses Member Activity.

Antarmuka Halaman Konfirmasi Proyek Petani



ripsi : Antarmuka halaman Konfirmasi Proyek Petani. Digunakan untuk mengkonfirmasi proyek yang dananya telah diterima oleh petani. Halaman ini muncul jika proyek yang dipilih pada List Proyek mempunyai status proyek "Menunggu Konfirmasi". Antarmuka berisi keterangan proyek yang dipilih, gambar bukti pendanaan yang diupload member dan button konfirmasi pendanaan.

: -

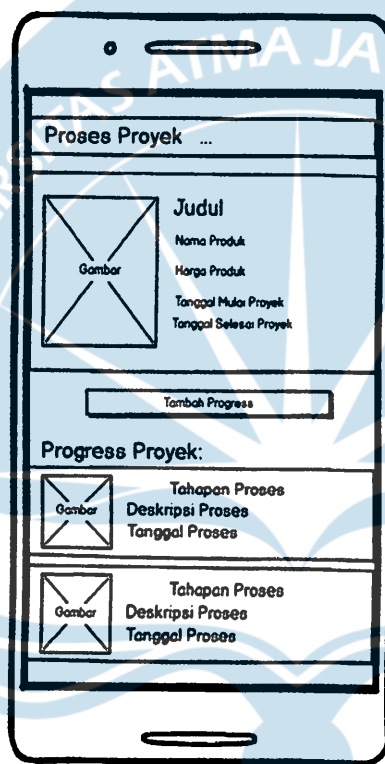
rt : Proyek sudah di konfirmasi.

od/Algoritma : **onClick() Button Konfirmasi**

1. Mengecek gambar sudah ada atau belum.
2. Jika gambar masih kosong muncul alert dialog "Belum Diupload Member".
3. Jika sudah maka memanggil prosedur update Status untuk mengupdate status proyek menjadi "Telah Didanai".
4. Prosedur update status mengirimkan request berisi data kode pendanaan dan status proyek yaitu 3.

5. API menerima request dan request ditangani oleh fungsi update status proyek.
6. Fungsi update status proyek mengubah status proyek berdasarkan kode pendanaan yang dikirim pada tabel pendanaan proyek dari status 2 menjadi status 3 yaitu menjadi "Telah didanai".
7. API mengirimkan response berupa status dan status message.
8. Jika status berhasil diubah maka tampilan beralih ke kelas Progresses Admin Activity.

## 20. Antarmuka Halaman List Proses Proyek Petani



- Deskripsi : Antarmuka halaman List Proses Proyek Petani. Antarmuka ini menampilkan proses dari proyek yang telah dipilih pada list proyek sebelumnya. Halaman ini muncul jika proyek yang dipilih pada List Proyek mempunyai status proyek "Telah Didanai". Antarmuka berisi keterangan proyek yang dipilih, list proses proyek yang telah petani buat dan button tambah proses.
- Input : -
- Output : List proses pada proyek tertentu.
- Method/Algoritma : **onCreate()**
1. Memanggil preference PREF\_PROGRESSSES untuk mendapatkan kode pendanaan.
  2. Memanggil prosedur showProgress() untuk melakukan request ke api.
  3. Api mengambil data – data berdasarkan kode pendanaan yang dikirim request dari tabel progress proyek. Api mengambil data kode progress, kode

pendanaan, tahapan progress, detail progress, gambar progress, dan lokasi progress.

4. Response di kirim dalam bentuk list progress proyek.

5. Response dikirim ke kelas Recycler List View Proses untuk dapat menampilkan hasil tersebut kedalam listview.

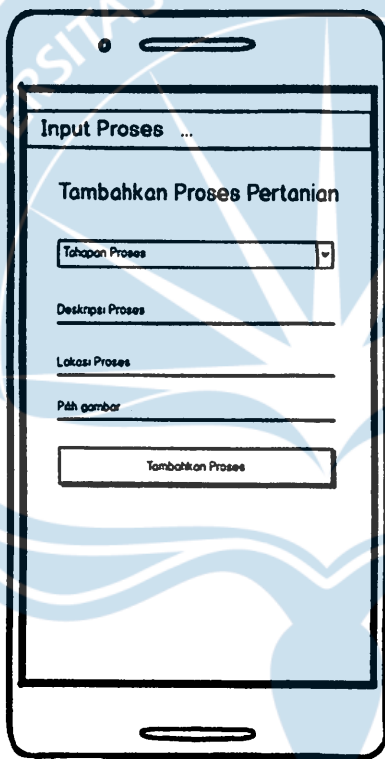
6. Dalam kelas Recycler List View Proses list response ditampilkan pada listview berdasarkan position/array list tersebut.

7. List response sudah ditampilkan ke halaman List Proses Proyek Petani.

#### **onClick Button Tambah Proses**

1. Tampilan berpindah ke kelas Input Progresses Activity.

#### Antarmuka Halaman Input Proses Proyek



psi : Antarmuka halaman Input Proses Proyek Petani. Antarmuka ini digunakan untuk memasukkan proses dari proyek yang telah dipilih pada list proyek sebelumnya. Antarmuka berisi beberapa data input dan button tambah proses.

- : • Semua field harus diisi
- : • Tanggal proses merupakan tanggal sekarang

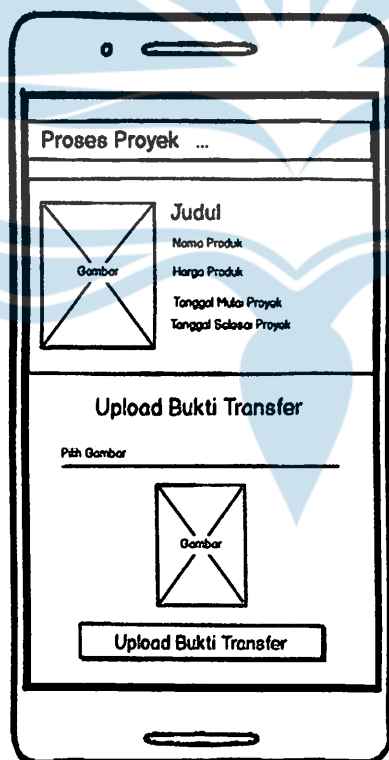
t : Data proses tersimpan.

d/Algoritma : **onClick Button Tambah**

1. Membaca data input.
2. Melakukan validasi data input.

3. Jika ada yang tidak valid, menampilkan peringatan dalam bentuk toast 'Data yang dimasukkan salah', kembali ke langkah 1.
4. Memanggil prosedur Input Proses Proyek untuk mengirimkan request. Data yang dikirim ke api ada kode pendanaan, tahapan proses, detail proses, lokasi proses, dan gambar.
5. Api menerima request dan request ditangani oleh fungsi post\_proses.
6. Fungsi post\_proses mengenerate kode proses dari fungsi generateKodeProses, kemudian memasukkan data - data ke tabel proses. Data yang dimasukkan kode proses, kode pendanaan, tanggal proses yang diambil dari sistem, tahapan proses, detail proses, gambar proses, dan lokasi proses.
7. Jika data berhasil disimpan, api mengirimkan response berupa status dan status message.
8. Prosedur Input Proses Proyek menerima response dan menampilkan status message dalam toast.
9. Tampilan berpindah ke Progresses Admin Activity / Halaman list proses proyek.

## 22. Antarmuka Halaman Upload Bukti Pendanaan



deskripsi : Antarmuka halaman Upload Bukti Pendanaan Proyek. Digunakan untuk menginputkan bukti pendanaan proyek yang didanai member. Halaman ini muncul jika proyek yang dipilih pada List Proyek mempunyai status proyek

“Menunggu Konfirmasi”. Antarmuka berisi keterangan proyek yang dipilih, button pilih gambar dan button upload bukti pendanaan.

Input : Gambar

Output : Gambar disimpan didatabase

Method/Algoritma : **onClick() Button Upload**

1. Menampilkan alert dialog, jika pengguna memilih “yes”, maka lanjut langkah 2.

2. Mengecek gambar sudah ada atau belum. Jika gambar masih kosong muncul alert dialog “Belum Diupload Member”.

3. Jika sudah maka memanggil prosedur update Bukti Transfer.

4. Prosedur update Bukti Transfer mengirimkan request berisi data kode detail transaksi dan gambar yang sudah di encoded ke base64.

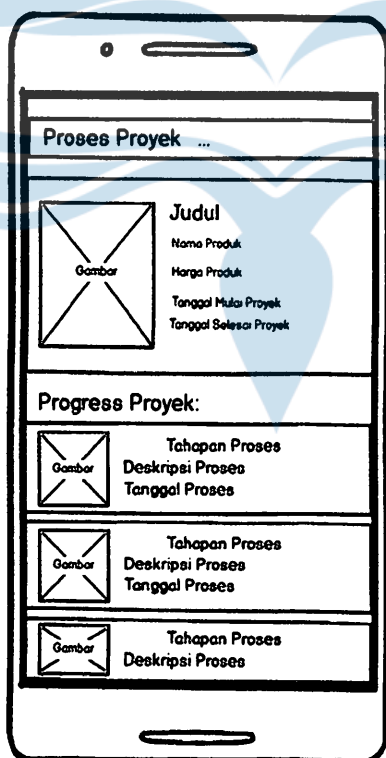
5. API menerima request dan request ditangani oleh fungsi update bukti transfer.

6. Fungsi update bukti transfer mengupdate bukti\_pendanaan pada tabel detail\_pendanaan berdasarkan kode detail transaksi yang dikirim.

7. API mengirimkan response berupa status dan status message.

8. Jika status berhasil diubah maka tampilan beralih ke kelas Progresses Member Fragment / List Proyek milik member.

### 23. Antarmuka Halaman List Proses Proyek Member



Deskripsi : Antarmuka halaman List Proses Proyek Member. Antarmuka ini menampilkan proses dari proyek yang telah dipilih pada list proyek

sebelumnya. Halaman ini muncul jika proyek yang dipilih pada List Proyek mempunyai status proyek "Telah Didanai". Antarmuka berisi keterangan proyek yang dipilih dan list proses proyek yang telah petani buat.

: -

: List proses pada proyek tertentu.

od/Algoritma : onCreate()

1. Memanggil preference PREF\_PROGRESSES untuk mendapatkan kode pendanaan.

2. Memanggil prosedur showProgress() untuk melakukan request ke api.

3. Api mengambil data – data berdasarkan kode pendanaan yang dikirim request dari tabel progress proyek. Api mengambil data kode progress, kode pendanaan, tahapan progress, detail progress, gambar progress, dan lokasi progress.

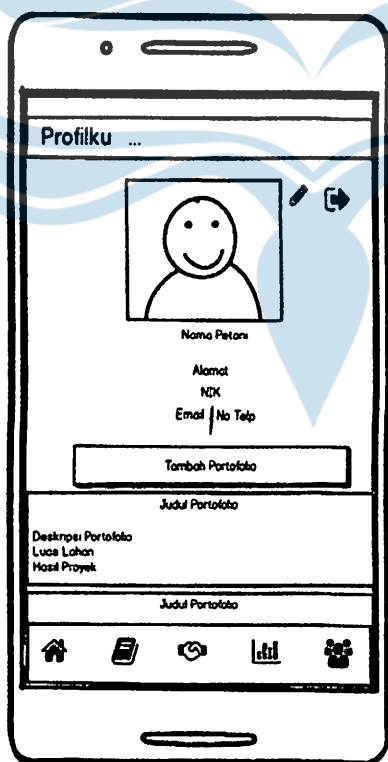
4. Response di kirim dalam bentuk list progress proyek.

5. Response dikirim ke kelas Recycler List View Proses untuk dapat menampilkan hasil tersebut kedalam listview.

6. Dalam kelas Recycler List View Proses list response ditampilkan pada listview berdasarkan position/array list tersebut.

7. List response sudah ditampilkan ke halaman List Proses Proyek member.

#### 4. Antarmuka Halaman Profil Petani



ripsi

: Antarmuka halaman Profil Petani. Antarmuka ini menampilkan profil petani yang sedang dalam aplikasi dan menampilkan portofolio milik petani pada

bagian bawah profil. Antarmuka berisi profil petani seperti foto profil, nama, alamat, no identitas, dan nomor telepon. Selain itu antarmuka ini berisi list portofolio milik petani, dan button tambah portofolio. Juga terdapat button edit profil petani dan button log out aplikasi.

Input : -

Output : Profil petani dan list portofolio.

Method/Algoritma : **onCreate()**

1. Memanggil preference untuk mendapatkan id admin.
2. Memanggil prosedur loadProfileAdmin() untuk melakukan request ke api.
3. Api mengambil data – data admin berdasarkan id admin yang dikirim request dari tabel admin petani. Api mengambil data id admin, user name, password, nama lengkap, no identitas, email, no telp, alamat, dan foto.
4. Response di kirim dalam bentuk data petani.
5. Response dimasukkan ke parameter yang sudah dideklarasikan sesuai dengan kebutuhan dan ditampilkan ke layout.
6. Response juga disimpan ke preference PREF\_ADMIN untuk dapat di tampilkan di layout edit profil.
7. Memanggil prosedur show Portofolio Admin. Prosedur ini untuk menampilkan list portofolio admin.
8. Prosedur showPortofolioAdmin mengirim request ke api dengan data yang dikirim adalah id admin.
9. Api menerima request dan request ditangani oleh fungsi getPortofolio,
10. Fungsi getPortofolio mengambil data dari tabel portofolio admin berdasarkan id admin yang dikirim. Data yang diambil kode portofolio, id admin, judul, deskripsi, luas lahan, dan hasil panen.
11. Api mengirimkan response berupa list portofolio.
12. Prosedur show Portofolio admin menerima response dan response dikirim ke kelas Recycler List View Portofolio Admin.
13. Dalam kelas Recycler List View Portofolio Admin list response ditampilkan pada listview berdasarkan position/array list tersebut.
14. List response sudah ditampilkan ke halaman Profil Petani.

#### **onClick Button Tambah Portofolio**

1. Tampilan berpindah ke kelas Input Portofolio Activity.

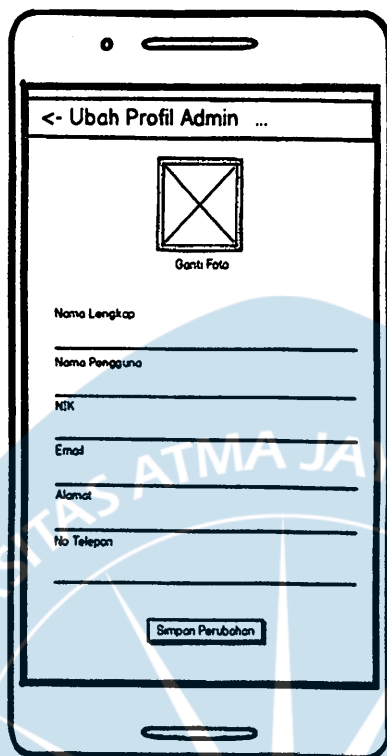
#### **onClick Button Edit Profil Petani**

1. Tampilan berpindah ke kelas Edit Profile Admin Activity.

#### **onClick Button Logout**

1. Muncul alert dialog konfirmasi ingin logout.
2. Jika pengguna memilih “Ya”, maka akan keluar dari aplikasi dan preference pengguna akan dihapus.

## 25. Antarmuka Halaman Edit Profil Petani



- deskripsi** : Antarmuka halaman Edit Profil Petani. Antarmuka ini mengubah profil petani yang sedang log In aplikasi. Antarmuka berisi data input dan button Simpan perubahan.
- output** :
  - Nama lengkap, Email, NIK, Username, Password tidak boleh kosong
  - Format email harus benar
  - NIK berupa angka 16 digit
  - Username minimal 6 karakter
- input** : Data profil terupdate.
- Method/Algoritma** : **onClick Button Simpan**
1. Membaca data input.
  2. Melakukan validasi data input.
  3. Jika ada yang tidak valid, menampilkan peringatan dalam bentuk toast 'Data yang dimasukkan salah', kembali ke langkah 1.
  4. Memanggil prosedur update Admin untuk mengirimkan request. Data yang dikirim ke api ada id admin, user name, nama lengkap, no identitas, email, alamat, no telepon dan foto.
  5. Api menerima request dan request ditangani oleh fungsi update\_admin.
  6. Fungsi update\_admin mengupdate tabel admin petani dengan data yang diupdate adalah username, nama lengkap, no identitas, email, alamat, no telepon, dan foto. Data yang diupdate yang memiliki id admin sama dengan id admin yang dikirim request.

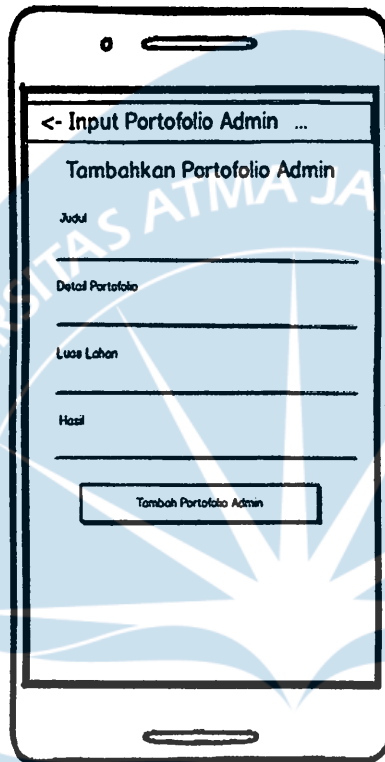


7. Jika data berhasil disimpan, api mengirimkan response berupa status dan status message.

8. Prosedur Update Admin menerima response dan menampilkan status message dalam toast.

9. Tampilan berpindah ke Profile Admin Fragment / Halaman profil admin.

## 26. Antarmuka Halaman Input Porfolio Petani



**Deskripsi** : Antarmuka halaman Input Portofolio Petani. Antarmuka ini menginputkan portofolio petani yang sedang log In aplikasi. Antarmuka berisi data input dan button tambah portofolio admin.

**Input** :

- Semua field harus diisi
- Luas lahan / jumlah ternak dalam satuan hektar / ekor
- Hasil panen / ternak dalam satuan kg / ekor

**Output** : Data portofolio baru tersimpan.

**Method/Algoritma** : **onClick Button Tambah**

1. Membaca data input.
2. Melakukan validasi data input.
3. Jika ada yang tidak valid, menampilkan peringatan dalam bentuk toast 'Data yang dimasukkan salah', kembali ke langkah 1.
4. Memanggil prosedur Input portofolio untuk mengirimkan request. Data yang dikirim ke api ada id admin, judul, deskripsi, luas lahan, dan hasil panen.
5. Api menerima request dan request ditangani oleh fungsi post\_portofolio.

6. Fungsi `post_portofolio` mengenerate kode portofolio dari fungsi `generateKodePortofolio`, kemudian memasukkan data - data ke tabel portofolio petani. Data yang dimasukkan kode portofolio, id admin, judul, deskripsi, luas lahan, dan hasil panen.
7. Jika data berhasil disimpan, api mengirimkan response berupa status dan status message.
8. Prosedur Input Portofolio menerima response dan menampilkan status message dalam toast.
9. Tampilan berpindah ke Profile Admin Fragment / Halaman profil admin.

## 27. Antarmuka Halaman Profil Member



- Deskripsi** : Antarmuka halaman Profil Member. Antarmuka ini menampilkan profil member yang sedang dalam aplikasi. Antarmuka berisi profil petani seperti foto profil, nama, alamat, no identitas, dan nomor telepon. Juga terdapat button edit profil petani dan button log out aplikasi.
- Input** : -
- Output** : Profil Member.
- Method/Algoritma** : `onCreate()`
1. Memanggil preference untuk mendapatkan id member.
  2. Memanggil prosedur `loadProfileMember()` untuk melakukan request ke api.

3. Api mengambil data – data admin berdasarkan id member yang dikirim request dari tabel member. Api mengambil data id member, user name, password, nama lengkap, no identitas, emali, no telp, alamat, dan foto.
4. Response di kirim dalam bentuk data member.
5. Response dimasukkan ke parameter yang sudah dideklarasikan sesuai dengan kebutuhan dan ditampilkan ke layout.
6. Response juga disimpan ke preference PREF\_MEMBER untuk dapat di tampilkan di layout edit profil.

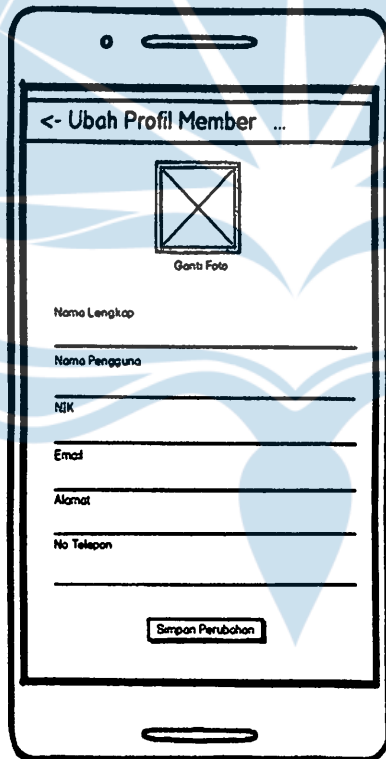
**onClick Button Edit Profil Member**

1. Tampilan berpindah ke kelas Edit Profile Member Activity.

**onClick Button Logout**

1. Muncul alert dialog konfirmasi ingin logout.
2. Jika pengguna memilih “Ya”, maka akan keluar dari aplikasi dan preference pengguna akan dihapus.

28. Antarmuka Edit Profil Member



**Deskripsi** : Antarmuka halaman Edit Profil Member. Antarmuka ini mengubah profil member yang sedang log In aplikasi. Antarmuka berisi data input dan button Simpan perubahan.

- Input** :
- Nama lengkap, Email, NIK, Username, Password tidak boleh kosong
  - Format email harus benar
  - NIK berupa angka 16 digit
  - Username minimal 6 karakter

Output	: Data profil terupdate.
Method/Algoritma	: <b>onClick Button Simpan</b> <ol style="list-style-type: none"> <li>1. Membaca data input.</li> <li>2. Melakukan validasi data input.</li> <li>3. Jika ada yang tidak valid, menampilkan peringatan dalam bentuk toast ' Data yang dimasukkan salah' , kembali ke langkah 1.</li> <li>4. Memanggil prosedur update Member untuk mengirimkan request. Data yang dikirim ke api ada id member, user name, nama lengkap, no identitas, email, alamat, no telepon dan foto.</li> <li>5. Api menerima request dan request ditangani oleh fungsi update_member.</li> <li>6. Fungsi update_member mengupdate tabel member dengan data yang diupdate adalah username, nama lengkap, no identitas, email, alamat, no telepon, dan foto. Data yang diupdate yang memiliki id member sama dengan id member yang dikirim request.</li> <li>7. Jika data berhasil disimpan, api mengirimkan response berupa status dan status message.</li> <li>8. Prosedur Update Member menerima response dan menampilkan status message dalam toast.</li> <li>9. Tampilan berpindah ke Profile Member Fragment / Halaman profil Member.</li> </ol>

