



## PENGEMBANGAN SISTEM JUAL BELI BAHAN POKOK PETANI BERBASIS APLIKASI MOBILE

Muhammad Hamas<sup>1</sup>, Zaki Imaduddin<sup>2</sup>

<sup>1,2</sup>Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri  
Jakarta Selatan, DKI Jakarta, Indonesia 12640  
[hamatsukida@gmail.com](mailto:hamatsukida@gmail.com) , [zaki\\_ip@yahoo.com](mailto:zaki_ip@yahoo.com)

### Abstract

*This final project discusses the planning of making a mobile application that is the application of buying and selling basic raw materials from farmers or can be called a self-sufficiency application, from the analysis to the stage of making the application to be made. It is expected that this application research can make an alternative in the process of buying and selling basic commodities directly from farmers so that there is price transparency. This application will be designed using android studio desktop tools and a database using Firebase.*

**Keywords:** Mobile Application, Swasembada, Android Studio and Firebase

### Abstrak

Tugas akhir ini membahas tentang perencanaan pembuatan aplikasi mobile yaitu aplikasi jual beli bahan baku pokok dari petani atau bisa disebut dengan aplikasi Swasembada, dari mulai analisis hingga sampai ke tahap pembuatan aplikasi yang akan di buat. Diharapkan dari adanya penelitian aplikasi ini dapat menjadikan salah satu alternatif dalam peroses jual beli bahan pokok langsung dari petani sehingga adanya transparansi harga. Aplikasi ini akan dirancang dengan menggunakan *tools desktop* Android Studio dan *database* menggunakan Firebase.

**Kata kunci:** Aplikasi Mobile, Swasembada, Android Studio dan Firebase

### 1. PENDAHULUAN

Indonesia adalah salah satu negara agraris terpenting di Asia Tenggara. Sebagai negara agraris, informasi sumber daya lahan memegang peranan yang sangat penting dalam mendukung pembangunan pertanian. Hal ini karena potensi sumber daya lahan terkait dengan berbagai aspek, baik teknis maupun sosial, ekonomi, hukum bahkan budaya, serta mempunyai peran strategis untuk kebanyakan sektor pembangunan lainnya.

Data BPS (2013) menyebutkan luas daratan Indonesia + 191,09 juta ha yang meliputi 17.000 pulau besar dan kecil. Luas daratan tersebut merupakan luasan terbesar untuk suatu negara kepulauan. Daratan ini dapat berupa lahan kering, lahan rawa, dan lahan basah non-rawa yang penggunaannya saat ini dapat berupa hutan, lahan pertanian, semak/belukar, permukiman, dan lainnya. Sejalan dengan makin bertambahnya jumlah penduduk dan terus menciutnya luas lahan pertanian akibat konversi lahan ke penggunaan lain, maka kebutuhan sumber daya lahan makin terus meningkat. Oleh sebab itu, informasi sumber daya lahan yang mencakup luas, sebaran dan potensi

ketersediaannya untuk pertanian semakin sangat diperlukan [1].

Pada dasarnya, sektor pertanian masih merupakan sumber pendapatan yang penting bagi masyarakat perdesaan [2]. Hasil panen petani baik dalam jumlah besar maupun kecil akan sangat menentukan keberlangsungan hidup mereka. Pada kenyataannya, tingkat kemiskinan dibeberapa daerah menunjukkan bahwa masyarakat miskin berada di pedesaan. Lebih dari itu, anggota keluarga petani juga memiliki level pendidikan yang rendah serta akses kesehatan yang minim [3].

Dari hasil penelitian informasi-informasi sebelumnya dengan ini penulis membuat dan merancang aplikasi mobile, dalam pembuatan perancangan aplikasi jual beli bahan baku pokok langsung dari petani dengan menggunakan aplikasi berjenis mobile yang dapat dengan mudah di install di smartphone atau handphone sehingga diharapkan mempermudah untuk adanya proses transaksi jual dan beli hasil dari pada panen para petani yang langsung di pasarkan kepada masyarakat melalui aplikasi

yang akan di buat oleh pengembang yaitu aplikasi jual beli bahan baku pokok dari petani berbasis mobile (swasembada).

### 1.1 Rumusan Masalah

Berdasarkan dari latar belakang sebelumnya dapat dirumuskan permasalahan sebagai berikut:

1. Apakah dengan adanya sistem aplikasi jual beli bahan baku, berbasis *mobile* dibutuhkan oleh masyarakat baik penjual dan pembeli?
2. Apakah aplikasi jual beli hasil panen, berbasis *mobile* dapat membantu menjual hasil panen dan membeli hasil panen?

### 1.2 Batasan Masalah

Oleh karena besarnya permasalahan dan keterbatasan waktu serta pengetahuan penulis maka agar pembahasan tidak menyimpang dari tujuan dilakukan pembatasan masalah sebagai berikut:

1. Penulis membatasi konsumen dari pelaku usaha yang memanfaatkan bahan baku dari petani.
2. Penulis merancang *system* aplikasi berbasis *mobile*, dengan menggunakan *android studio*, dan *database firebase*.

### 1.3 Tujuan dan Manfaat

Apabila penelitian dan pembuatan aplikasi ini berhasil dengan baik, diharapkan dapat memberikan manfaat sebagai berikut:

#### 1.3.1 Bagi Institusi

Penelitian ini diharapkan berguna sebagai tolak ukur dan referensi institusi pendidikan dalam melakukan penelitian berbasis produk aplikasi atau jual beli yang bermanfaat untuk petani.

#### 1.3.2 Bagi Masyarakat

Memberikan sumbangan kepada masyarakat luas untuk dapat meningkatkan taraf hidup masyarakat serta mendorong untuk mandiri secara ekonomi dan lebih memudahkan dalam memudahkan pembelian bahan baku pokok langsung dari petani.

#### 1.3.3 Bagi Petani

Dapat memudahkan parapetani untuk penjualan hasil panen atau bahan baku pokok makanan langsung ke pelanggan atau masyarakat.

#### 1.3.4 Bagi Penulis

Menambah wawasan dari sistem pertanian terutama di Indonesia dan memahami dari sistem pemasarannya.

## 2. TINJAUAN PUSTAKA

### 2.1 UML (*Unified Modelling Language*)

*Unified Modeling Language (UML)* adalah sebuah bahasa pemodelan grafis yang digunakan sebagai standar untuk memodelkan system dengan metodologi pemodelan berorientasi objek.

*Unified Modeling Language (UML)* di standarkan oleh *Object Management Group (OMG)*. *UML* pertamakali dipopulerkan oleh Grady Booch dan James Rumbaugh pada akhir tahun 1994. Kemudian Ivar Jacobos yang merupakan pimpinan dari *Object Oriented Development* bergabung. Pada tahun 1996 *UML* mulai digunakan sebagai *tools* untuk memodelkan *system* pada IBM dan i-logix [4].

Berikut adalah 4 (empat) fungsi dan kegunaan dari *UML* yaitu:

1. *Virtualizing*
2. *Specifying*
3. *Constructing*
4. *Documenting*

*Virtualizing* adalah sebagai alat komunikasi atau konsep model antara tim pengembang system ( system analis dengan programmer).

*Specifying* adalah sebagai *tools* yang digunakan untuk memodelkan system secara tepat dan jelas.

*Constructing* adalah *UML* sebagai bahasa grafis (bukan bahasa pemograman) mampu melakukan *mapping* dari konsep model kedalam bahasa pemograman.

*Documenting* adalah *UML* digunakan sebagai *tools* untuk melakukan dokumentasi teknis sebuah *system*.

Diagram-diagram yang terdapat pada *UML* sebagai berikut:

1. *Use Case Model*
2. *Class Diagram*
3. *Object Diagram*
4. *Activity Diagram*
5. *Sequence Diagram*
6. *Collaboration Diagram*
7. *Statechart Diagram*
8. *Component Diagram*
9. *Deployment Diagram*

Berikut adalah penjelasan dari jenis-jenis pemodelan *UML* diagram diatas yaitu:

1. *Use Case Model*  
*Use case model* merupakan kumpulan diagram dan text yang saling bekerja sama untuk mendokumentasikan bagaimana *user* atau *actor* berinteraksi dengan *system*, *use case model* terdiri dari beberapa diagram yaitu:

- a. *Use case diagram* adalah untuk menjelaskan konsep hubungan antara system dengan dunia luar, atau menggambarkan dan merepresentasikan actor, *use case*, dan dependencies.
  - b. *Use case narrative* adalah deskripsi yang menjelaskan *use case diagram*.
  - c. *Use case scenario* adalah pemecahan kemungkinan logika pada *use case diagram*.
2. *Class Diagram*  
Class diagram adalah diagram yang digunakan untuk merepresentasikan kelas, komponen-komponen kelas dan hubungan masing-masing kelas, mendeskripsikan jenis-jenis objek dalam sistem.
  3. *Object Diagram*  
*Object diagram* adalah diagram yang digunakan untuk menggambarkan dan merepresentasikan objek dan hubungan antar objek tersebut.
  4. *Activity Diagram*  
*Activity diagram* adalah diagram UML yang digunakan untuk menggambarkan alur aktivitas dari suatu proses. *Activity diagram* ini sangat mirip dengan *flowchart*.
  5. *Sequence Diagram*  
*Sequence diagram* adalah diagram yang menggambarkan interaksi antar objek. *Sequence diagram* secara khusus menjabarkan *behavior* sebuah skenario tunggal.
  6. *Collaboration Diagram*  
*Collaboration diagram* hampir sama dengan *sequence diagram*, dapat dikatakan *collaboration diagram* merupakan pengganti *sequence diagram*.
  7. *Statechart Diagram*  
*Statechart Diagram* adalah teknik umum yang digunakan untuk menggambarkan *behavior* sebuah *system*.
  8. *Component Diagram*  
*Component diagram* adalah diagram yang menggambarkan dan merepresentasikan model fisik dari implementasi *system software*.
  9. *Deployment Diagram*  
Deployment diagram adalah diagram yang merepresentasikan model fisik dari hardware serta integrasi dan distribusi software pada arsitektur hardware tersebut.

## 2.2 Metode Waterfall

Metode *Waterfall* adalah salah satu metode dalam SDLC yang mempunyai ciri khas pengerjaan yaitu setiap fase dalam *Waterfall* harus diselesaikan terlebih dahulu sebelum melanjutkan ke fase selanjutnya. Artinya focus terhadap masing-masing fase dapat dilakukan maksimal karena jarang adanya pengerjaan yang sifatnya parallel walaupun dapat saja terjadi paralelisme dalam *Waterfall* [5].

Fase dalam metode *Waterfall*, tahapan-tahapan dari metode *Waterfall* adalah sebagai berikut:

1. *Requirement Analysis*  
Seluruh kebutuhan *software* harus bisa didapatkan dalam fase ini termasuk didalamnya kegunaan *software* yang diharapkan penggunaan dan batasan *software*. Informasi ini biasanya dapat diperoleh melalui wawancara, survei atau diskusi.
2. *System Design*  
Tahap ini dilakukan sebelum melakukan ke proses *coding* atau pemrograman. Tahap ini bertujuan untuk membuat *design* dari hasil analisis yang telah didapat sesuai dengan kebutuhan *hardware* dan *system* secara keseluruhan.
3. *Implementation*  
Ditahap ini dilakukannya proses *coding* atau pemrograman. Pembuatan *software* dipecah menjadi modul-modul kecil yang nantinya akan digabungkan dalam tahap berikutnya.
4. *Integration and Testing*  
Ditahap ini dilakukannya penggabungan modul-modul yang sudah dibuat dan dilakukan pengujian ini dilakukan untuk mengetahui apakah *software* yang dibuat telah sesuai dengan *design* yang telah dibuat sebelumnya.
5. *Operation and Maintenance*  
Tahap ini adalah proses dilakukannya pemeliharaan, seperti kemungkinan adanya *bugs*, *error*, dan malfungsi terhadap *software* aplikasi yang telah dibuat.

## 2.3 Android Studio

Android Studio adalah Lingkungan Pengembangan Terpadu-Integrated Development Environment (IDE) untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA. Selain merupakan editor kode IntelliJ dan alat pengembang yang berdaya guna, Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas Anda saat membuat aplikasi Android, misalnya:

- a. Sistem versi berbasis Gradle yang fleksibel.
- b. Emulator yang cepat dan kaya fitur.
- c. Lingkungan yang menyatu untuk pengembangan bagi semua perangkat Android.
- d. Instant Run untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru.
- e. Template kode dan integrasi GitHub untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh.
- f. Alat pengujian dan kerangka kerja yang ekstensif.
- g. Alat Lint untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain.
- h. Dukungan C++ dan NDK.
- i. Dukungan bawaan untuk Google Cloud Platform, mempermudah pengintegrasian Google Cloud

## Messaging dan App Engine.

Laman ini berisi pengantar dasar fitur-fitur Android Studio. Untuk memperoleh rangkuman perubahan terbaru, lihat Catatan Rilis Android Studio [6].

### 2.4 Struktur Proyek

Setiap proyek di Android Studio berisi satu atau beberapa modul dengan file kode sumber dan file sumber daya. Jenis-jenis modul mencakup:

- a. Modul Aplikasi Android
- b. Modul Pustaka
- c. Modul Google App Engine

Secara *default*, Android Studio akan menampilkan file proyek Anda dalam tampilan proyek Android, seperti yang ditampilkan dalam gambar 1. Tampilan disusun berdasarkan modul untuk memberikan akses cepat ke file sumber utama proyek Anda.

Semua file versi terlihat di bagian atas di bawah Gradle Scripts dan masing-masing modul aplikasi berisi folder berikut:

1. Manifests:  
Berisi file `AndroidManifest.xml`.
2. Java:  
Berisi file kode sumber Java, termasuk kode pengujian JUnit.
3. Res:  
Berisi semua sumber daya bukan kode, seperti tata letak XML, string UI, dan gambar bitmap.

Struktur proyek Android pada disk berbeda dari representasi rata ini. Untuk melihat struktur file sebenarnya dari proyek ini, pilih Project dari menu tarik turun Project (dalam gambar 1), struktur ditampilkan sebagai Android) [6].

### 2.5 Firebase

Firebase adalah BaaS (*Backend as a Service*) yang saat ini dimiliki oleh Google. Firebase ini merupakan solusi yang ditawarkan oleh Google untuk mempermudah pekerjaan *Mobile Apps Developer*. Dengan adanya Firebase, *apps developer* bisa fokus mengembangkan aplikasi tanpa harus memberikan *effort* yang besar untuk urusan *backend* [3]. Dengan pengertian di atas penulis akan menggunakan Firebase untuk menghubungkan data-data informasi dari aplikasi yang di buat dan di hubungkan ke Firebase.

Berikut adalah beberapa fitur dari Firebase yang digunakan untuk membuat aplikasi Swasembada dan penjelasannya:

- a. Firebase Authentication  
Sebagian besar aplikasi perlu mengetahui identitas pengguna. Dengan mengetahui identitas pengguna, aplikasi dapat menyimpan data pengguna secara aman

di *cloud* dan memberikan pengalaman personal yang sama di setiap perangkat pengguna [7].

- b. *Cloud Storage*  
*Cloud Storage for Firebase* adalah layanan penyimpanan objek yang andal, sederhana, dan hemat biaya yang dibuat untuk skala *Google*. Firebase SDK untuk *Cloud Storage* menambahkan keamanan *Google* pada *upload* dan *download file* untuk aplikasi yang akan di buat [8].
- c. *Firebase Realtime Database*  
*Firebase Realtime Database* adalah *database* yang di-host di *cloud*. Data disimpan sebagai JSON dan disinkronkan secara realtime ke setiap klien yang terhubung, dengan menggunakan ini semua klien akan berbagi sebuah *instance realtime database* dan menerima *update* data terbaru secara otomatis [9].

## 3. METODOLOGI PENELITIAN

### 3.1 Alur Tahapan Penelitian

Alur tahapan penelitian digunakan untuk memudahkan penulis untuk merancang aplikasi yang dibuat, berikut:

SDLC adalah sebuah metode yang digunakan untuk mengembangkan sebuah system dan sebuah proses logika yang digunakan oleh seorang *system analyst* untuk mengembangkan sebuah system informasi yang melibatkan *requirement, validation, training* dan pemilik *system*.

SDLC identik dengan teknik pengembangan *system waterfall*, karena tahapannya menurun dari atas ke bawah. Berikut tahapan dari SDLC:

1. Planning  
Adanya perencanaan sebagai tahap awal untuk melakukan sebuah penelitian sebelum system dibuat dan dikembangkan seperti adanya pengumpulan data-data yang real, akurat, valid dan reliabel dengan menggunakan system wawancara, observasi dan studi literatur.
2. Analisis  
Setelah planning dan pengumpulan data, kemudian adanya proses analisis, seperti menganalisis system apa yang sangat dibutuhkan para petani, baik penjual dan pembeli. Dengan ini penulis menganalisis dan dapat memberikan solusi dalam prose jual beli atau transaksi antara petani dengan konsumen atau masyarakat.
3. Design  
Setelah menganalisis dan perencanaan penulis membuat desain aplikasi transaksi jual beli bahan baku antara petani dan pembeli atau konsumen, dari tampilan login, register, menu home, menu transaksi, menu penjualan dan masih banyak lagi. Untuk lebih jelasnya akan dibahas di bab IV.
4. Implementasi Sistem (Kode Program)

Setelah proses perencanaan, analisis, dan *design*. Selanjutnya yaitu proses implementasi dan algoritma kode pemrograman yang akan di buat sesuai design yang telah dibuat, bahasa pemrograman yang akan digunakan yaitu *Java* yang sudah tersedia di Android Studio 2.3.3 dan untuk *database* menggunakan *Firestore*.

#### 5. Pengujian

Setelah proses pembuatan *software application* telah dibuat, maka dilakukannya pengujian aplikasi berupa *software apk* yang sudah di *install* dan melakukan pengujian terhadap *system* yang telah dibuat dari mulai proses *user interface*, *input*, *output* dan *edit* pada aplikasi yang dibuat, dan aplikasi ini akan di uji terlebih dahulu oleh mahasiswa STT NF, hal ini dilakukan untuk memastikan keberhasilan sebuah aplikasi yang telah dibuat.

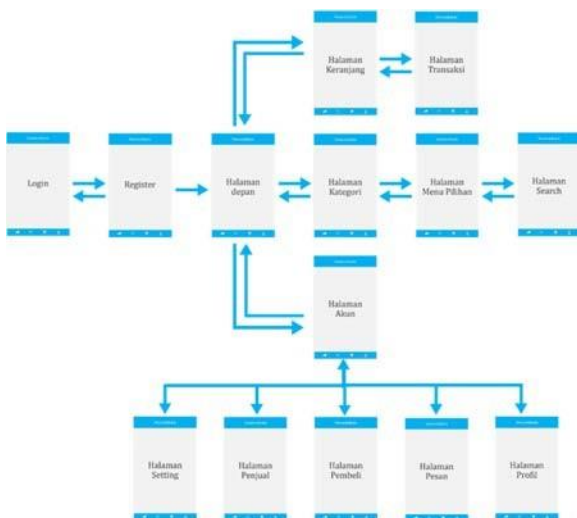
#### 3.2 Alat dan Bahan

Adapun *software* dan *hardware* yang digunakan untuk merancang aplikasi yang akan di buat antara lain:

- Software* dan *hardware* yang akan digunakan:
  - Laptop Dell Inspiron 15 5000 Series
  - Windows 10
  - Smartphone Samsung Galaxy J7
  - Android studio 3.2.3
- Bahasa Pemrograman:
  - Java, C, XML
- Database:
  - Firestore

#### 4. PERANCANGAN APLIKASI

Pada perancangan aplikasi ini akan menjelaskan atau menunjukkan *wireframe* yaitu rancangan awal sebuah desain pada aplikasi swasembada atau merupakan gambar model atau prototif halaman aplikasi *mobile* secara menyeluruh dan detail.



Gambar 1. Perancangan Wireframe Aplikasi Swasembada

#### 5. IMPLEMENTASI DAN PENGUJIAN

##### 5.1 Implementasi

Aplikasi jual beli bahan baku pokok yang langsung dari petani dan peternak yang dijual melalui aplikasi Swasembada yaitu aplikasi yang menjual hasil panen para petani ke masyarakat seperti pembisnis rumah makan, dan kebutuhan sehari-hari ibu rumah tangga. Aplikasi swasembada ini dirancang dengan *native application development* untuk penggunaan perangkat Android yang dimana aplikasi swasembada ini dibuat menggunakan *tools* atau aplikasi desktop yaitu Android Studio. Berikut adalah contoh gambaran implementasi alur rancangan aktifitas yang terhubung dengan aplikasi Swasembada.



Gambar 2. Arsitektur Implementasi Rancangan Kegiatan Aplikasi Swasembada

Android Studio adalah salah satu aplikasi *desktop* yang digunakan untuk membuat aplikasi berbasis *mobile* Android. Android studio ini menggunakan bahasa pemrograman Java untuk mendukung dalam pembuatan aplikasi *mobile* seperti aplikasi Swasembada yang akan di buat dengan menggunakan bahasa pemrograman Java. Selain itu Android Studio menyediakan *platform* *Firestore database realtime* yang disediakan Google untuk media penyimpanan data berbasis *realtime*.

Aplikasi Swasembada ini menggunakan *platform database* *Firestore* yang digunakan untuk penyimpanan data seperti data *login*, data barang, data penjual, dan data *user*.

##### 5.1.1 Batasan Implementasi

Berikut penulis merancang beberapa batasan dalam pengimplementasian penelitian perancangan aplikasi Swasembada.

- Aplikasi Swasembada ini akan di-*install* dengan versi apk yang digunakan menggunakan mobile Android, dengan *minimum system operasi* 5.0 (Lollipop).
- Aplikasi Swasembada ini dirancang dengan target pengguna masyarakat umum dan petani.

##### 5.1.2 Implementasi Android Studio

Berikut adalah implementasi tahapan instalasi dalam penggunaan aplikasi atau *tools* android studio yang digunakan untuk membuat aplikasi swasembada. Perancang

di sini menggunakan laptop dengan *system operating* Windows 10.

### 1. *System requirement* Android Studio

- a) Windows 7/8/10 (32bit or 64bit).
- b) *Minimum* 3Gb Ram, 8Gb *recommended*.
- c) JDK 8.
- d) 1280 x 800 *minimum screen resolution*.

### 2. Instalasi Android Studio

- a) Luncurkan atau install file .exe android studio yang telah di download.
- b) Ikuti wizard persiapan untuk memasang Android Studio dan setiap alat SDK yang diperlukan.
- c) Jika pada sebagian sistem Windows, skrip peluncur tidak menemukan tempat JDK dipasang.
- d) Atur variable dengan cara. Pilih menu Start > Computer > System Properties > Advanced System Properties. Lalu buka tab Advanced > Environment Variables dan tambahkan variabel sistem yang baru JAVA\_HOME yang menunjuk ke folder JDK Anda, misalnya C:\Program Files\Java\jdk1.8.0\_77 [10].

### 3. Membuat *project* baru

- a) Buka aplikasi Android Studio.
- b) Pada tampilan awal klik '*new project*' lalu tambahkan nama *project* contoh '*Aplikasi Swasembada*' kemudian klik OK dan ikuti perintah selanjutnya.
- c) Kemudian membuat *main activity* Java dan *Fragment*, sesuai kebutuhan yang akan digunakan untuk membuat aplikasi Swasembada.

#### 5.1.3 Implementasi Firebase Android Studio

Berikut adalah implementasi proses instalasi Firebase di dalam Android Studio yang sebelumnya telah di-install.

1. Persyaratan untuk device
  - a) Perangkat yang menjalankan sistem operasi Android 4.0 (Ice Cream Sandwich) atau yang lebih baru dan layanan Google Play 15.0.0 atau yang lebih tinggi.
  - b) Android Studio versi terbaru atau *ter-update*.
2. Menggunakan *Firebase Assistant*
  - a) Klik *tools* di dalam Android Studio> klik *Firebase* untuk membuka jendela *assistant*.
  - b) Kemudian klik *Connect to Firebase* untuk menghubungkan ke *assistant*.
  - c) Tambahkan *google-service.json*.

#### 5.1.4 Implementasi Login with Google dengan Firebase

Berikut adalah proses implementasi autentikasi login dengan menggunakan google sign in di dalam android studio dengan menggunakan firebase. Sebelum memulai ada beberapa tutorial sebagai berikut:

- 1) Tambahkan *Firebase* ke dalam *project*.
- 2) Tambahkan dependensi untuk *Firebase Authentication* dan *login* dengan Google ke file '*build.gradle*' tingkat aplikasi Anda:
  - a) implementation 'com.google.firebase:firebase-auth:16.0.3'
  - b) implementation 'com.google.android.gms:play-services-auth:16.0.0'
- 3) Aktifkan *login* dengan google di *Firebase Console* [11].

#### 5.1.5 Implementasi Instalasi Aplikasi Swasembada

Berikut adalah tahapan *installasi* pada aplikasi Swasembada pada perangkat Android atau *smartphone* Android.

- 1) Gunakan kabel data USB dan hubungkan dengan komputer atau laptop dengan keadaan aplikasi *tools* Android Studio sedang dibuka.
- 2) Aktifkan *debugging* pada *smartphone* Anda dan hubungkan dengan kabel data *USB*.
- 3) Kemudian hubungkan dengan meng-klik salah satu *button* atau perintah untuk menginstall aplikasi *apk* Swasembada di dalam Android Studio.

#### 5.2 Evaluasi Pengujian

Pada tahapan ini akan dilakukan pengujian aplikasi swasembada yang telah di install dan pengujian dengan menggunakan metode *blackbox testing* dan metode *usability testing*. pada tahap pengujian menggunakan metode *blackbox testing* ini akan dilakukan pengujian oleh penulis sendiri sedangkan pada tahap pengujian *usability testing* akan di lakukan pengujian oleh masyarakat umum, seperti masyarakat yang sering membeli bahan baku pokok seperti beras, sayuran dan buah-buahan. dan masyarakat yang sering bercocok tanam seperti menanam sayuran dan buah-buahan.

#### 5.2.1 Rancangan Pengujian dengan *User Acceptance Test*

Pada proses evaluasi ini dilakukan proses pengujian terhadap aplikasi yang telah dikembangkan untuk penelitian ini. Proses uji yang digunakan yaitu dengan proses *User Acceptance Test* (UAT) dimana pada tahap ini dibuat daftar skenario pengujian yang nantinya akan langsung dipraktekan oleh *user*, proses ini untuk memastikan bahwa requirement user yang telah diimplementasikan sudah sesuai dengan yang diharapkan *user*.

## 6. KESIMPULAN DAN SARAN

### 6.1 Kesimpulan

Berdasarkan hasil dari analisis hingga perancangan *system* dan pengujian dengan *blackbox* yang telah dilakukan pengujian aplikasi *mobile* 'Swasembada' yaitu aplikasi jual beli bahan baku kebutuhan pokok masyarakat seperti beras, sayuran dan buah-buahan yang akan disimpulkan sebagai berikut.

1. Berdasarkan dari data informasi yang telah di buat dengan menggunakan metode kuisisioner menggunakan *google form* untuk mengetahui apakah aplikasi swasembada dibutuhkan oleh masyarakat, hasilnya dengan adanya aplikasi ini masyarakat ingin mencoba dan menggunakan aplikasi swasembada dan dapat membantu masyarakat dalam melakukan pembelian dan penjualan.
2. Dari hasil wawancara kepada petani sebagai penjual dan pembeli dengan menggunakan metode *User Acceptance Test* (UAT) bahwa aplikasi Swasembada dapat digunakan untuk penjualan dan pembelian dengan mudah.

## 6.2 Saran

Berdasarkan penelitian dan pembuatan aplikasi ini terdapat beberapa fitur-fitur belum sempurna dengan baik atau belum terselesaikan. Oleh karena itu penulis menyarankan untuk pengembangan penelitian yang selanjutnya untuk menyempurnakan fitur-fitur yang belum sempurna dengan baik antara lain:

1. Sempurnakan proses transaksi pada pembelian agar dapat terlihat proses dengan sebaik-baiknya dan berikan keamanan pada data pembelian.
2. Sempurnakan jumlah upload image lebih dari satu dan tambahkan beberapa detail pada deskripsi penjualan.

## DAFTAR PUSTAKA

- [1] Ritung et al., "Sumber Daya Lahan Penelitian Indonesia: Luas, Penyebaran dan Potensi," Laporan Teknis Balai Penelitian dan Pengembangan Sumberdaya Lahan Pertanian, 2015.
- [2] Kurniati and E. Dwi, "Pengaruh Kesejahteraan, Inovasi, dan Risiko terhadap Keputusan Petani Berwirausaha pada Sektor Industri di Wilayah Pedesaan Kabupaten Semarang," *Jurnal Ilmiah Inkoma*, Vol. 22, No. 1, 2011.
- [3] A. Paranata, et al., "Mengurai Model Kesejahteraan Petani," *Jurnal Ekonomi dan Kebijakan*, 2011
- [4] Prof. Dr. Sri Mulyani, Ak., CA. "Metode Analisis dan Perancangan Sistem," Page 24, [Online]. Available: <https://books.google.co.id/books?isbn=979199062X> [diakses 25 desember 2017]
- [5] S Janner, "Rekayasa Perangkat Lunak," Yogyakarta: Penerbit Andi, 2010.
- [6] Android Studio." [Online].<https://developer.android.com/studio/intro/index.html?hl=id> [diakses 13 November 2017]
- [7] "Firebase Autentication." [Online]. <https://firebase.google.com/docs/auth/> [diakses 07 Agustus 2019]
- [8] "Cloud Storage," [Online]. Available: <https://firebase.google.com/docs/storage/?hl=id> [diakses 07 Agustus 2019]
- [9] "Firebase Realtime Database," [Online]. Available: <https://firebase.google.com/docs/database/?hl=id> [diakses 07 Agustus 2019]
- [10] "Install Android Studio for Windows," [Online]. Available: <https://developer.android.com/studio/install?hl=id> [diakses 13 November 2017]
- [11] Melakukan Autentikasi menggunakan Login dengan Google Android" [Online]. Available: <https://firebase.google.com/docs/auth/android/google-signin?hl=i>