

**PENGEMBANGAN APLIKASI PELAPORAN HARGA
KOMODITAS PERTANIAN BERBASIS *MOBILE*
MENGUNAKAN REST API**

DIEVA SETYATAMA



**DEPARTEMEN ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
INSTITUT PERTANIAN BOGOR
BOGOR
2016**

PERNYATAAN MENGENAI SKRIPSI DAN SUMBER INFORMASI SERTA PELIMPAHAN HAK CIPTA

Dengan ini saya menyatakan bahwa skripsi berjudul Pengembangan Aplikasi Pelaporan Harga Komoditas Pertanian Berbasis *Mobile* Menggunakan REST API adalah benar karya saya dengan arahan dari komisi pembimbing dan belum diajukan dalam bentuk apa pun kepada perguruan tinggi mana pun. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam Daftar Pustaka di bagian akhir skripsi ini.

Dengan ini saya melimpahkan hak cipta dari karya tulis saya kepada Institut Pertanian Bogor.

Bogor, Agustus 2016

Dieva Setyatama
NIM G64120012

ABSTRAK

DIEVA SETYATAMA. Pengembangan Aplikasi Pelaporan Harga Komoditas Pertanian Berbasis *Mobile* Menggunakan REST API. Dibimbing oleh DEAN APRIANA RAMADHAN.

Sektor pertanian merupakan sektor yang berpengaruh dan mendominasi perekonomian Indonesia, terutama pada komoditas bawang, beras, cabai, kedelai, dan daging sapi. Fluktuasi harga dan keterbatasan informasi harga komoditas di pasar merupakan salah satu permasalahan dalam pertanian. Perkembangan teknologi *smartphone* dan internet saat ini dapat dimanfaatkan untuk membantu mengurangi permasalahan tersebut. Penelitian ini mengembangkan aplikasi pelaporan harga komoditas pertanian berbasis *mobile* menggunakan REST API yang mampu memberikan informasi harga komoditas secara langsung beserta lokasinya dari Google Maps. Metode pengembangan yang diterapkan dalam pengembangan aplikasi ini adalah *extreme programming*. Aplikasi ini mampu menyediakan media komunikasi berupa aspirasi petani dan opini masyarakat untuk pemerintah.

Kata kunci: aplikasi *mobile*, *extreme programming*, pertanian, REST API

ABSTRACT

DIEVA SETYATAMA. Development of Mobile-Based Agricultural Commodity Price Reporting Application Using REST API. Supervised by DEAN APRIANA RAMADHAN.

The agricultural sector is the most influential sector and dominate the economies of Indonesia, especially for onion, rice, chili, soy, and meat commodities. Price fluctuations and limited information of commodity prices is one of the problems in agriculture. The development of the internet and smartphone technology can be used to reduce the problem. This study developed a mobile-based reporting application in agricultural commodity prices using REST API that can provide information on commodity prices directly on Google Maps location. The development method that is applied in the development of this application is *extreme programming*. This application is also capable of providing communication media such as the aspirations of farmers and the public opinion to the government.

Keywords: agriculture, *extreme programming*, mobile application, REST API

**PENGEMBANGAN APLIKASI PELAPORAN HARGA
KOMODITAS PERTANIAN BERBASIS *MOBILE*
MENGUNAKAN REST API**

DIEVA SETYATAMA

Skripsi
sebagai salah satu syarat untuk memperoleh gelar
Sarjana Ilmu Komputer
pada
Departemen Ilmu Komputer

**DEPARTEMEN ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
INSTITUT PERTANIAN BOGOR
BOGOR
2016**

Penguji:

- 1 Ir Meuthia Rachmaniah, MSc
- 2 Dr Yani Nurhadriyani, SSi MT

Judul Skripsi: Pengembangan Aplikasi Pelaporan Harga Komoditas Pertanian
Berbasis *Mobile* Menggunakan REST API

Nama : Dieva Setyatama

NIM : G64120012

Disetujui oleh

Dean Apriana Ramadhan, SKomp MKom
Pembimbing

Diketahui oleh

Dr Ir Agus Buono, MSi MKom
Ketua Departemen

Tanggal Lulus:

PRAKATA

Puji dan syukur penulis panjatkan kepada Allah *subhanahu wa ta'ala* atas segala karunia-Nya, sehingga penulis dapat menyelesaikan tugas akhir dengan judul Pengembangan Aplikasi Pelaporan Harga Komoditas Pertanian berbasis *Mobile* Menggunakan REST API. Penelitian ini dilaksanakan mulai Maret sampai Juli 2016 bertempat di Departemen Ilmu Komputer FMIPA IPB.

Dalam pelaksanaan tugas akhir ini banyak pihak yang selalu memberikan dukungan dan bantuan. Oleh karena itu penulis ingin menyampaikan terima kasih yang sebesar-besarnya kepada:

- 1 Orang tua, kakak, dan adik penulis yang selalu memberikan doa, dukungan, kasih sayang, dan semangat yang tiada tara.
- 2 Bapak Dean Apriana Ramadhan, SKomp MKom selaku dosen pembimbing yang selalu memberikan bimbingan dan saran dengan kesabarannya selama pengerjaan tugas akhir.
- 3 Ibu Ir Meuthia Rachmaniah, MSc dan Ibu Dr Yani Nurhadriyani, SSi MT selaku dosen penguji yang telah memberikan kritik dan sarannya yang membangun dalam penelitian penulis.
- 4 Teman-teman satu bimbingan, yaitu Reza Fahlevi, Dian Setiana, Kadek Grihadevi, Indriyani, Moh Reza, dan Shellafuri Biru yang selalu memberikan kritik, saran, dukungan, dan semangatnya.
- 5 Ardhi, Kika, Ilham A, dan Rakha sebagai teman diskusi yang selalu memberikan bantuan, saran, dan kritiknya yang membangun dalam pengerjaan aplikasi.
- 6 Teman-teman Full House 102, yaitu Kartika Andiani, Kurnia Pangastuti, Nitya Fedita, Aprilita Putri, Retno Puspita, Wahyu Mukti, Ulvia Muspita, dan Hilma Alfiani yang selalu memberikan doa, motivasi, saran, dan semangatnya untuk penulis.
- 7 Teman-teman Vector 49 atas segala bantuan, kebersamaan, ilmu, informasi, dan semangat yang menunjang dalam penyelesaian tugas akhir ini.

Semoga karya ilmiah ini bermanfaat.

Bogor, Agustus 2016

Dieva Setyatama

DAFTAR ISI

DAFTAR TABEL	vi
DAFTAR GAMBAR	vi
DAFTAR LAMPIRAN	vi
PENDAHULUAN	1
Latar Belakang	1
Perumusan Masalah	2
Tujuan Penelitian	2
Manfaat Penelitian	2
Ruang Lingkup Penelitian	3
TINJAUAN PUSTAKA	3
<i>Application Programming Interface</i> (API)	3
<i>Representational State Transfer</i> (REST)	3
<i>Roadmap</i> Penelitian	4
Undang-Undang RI Nomor 19 Tahun 2013	5
METODE PENELITIAN	5
Data Penelitian	5
Perancangan Arsitektur Penelitian	5
Tahapan Penelitian	6
Lingkungan Pengembangan	7
HASIL DAN PEMBAHASAN	7
Data Penelitian	7
Arsitektur Penelitian	8
Tahapan Penelitian	9
SIMPULAN DAN SARAN	20
Simpulan	20
Saran	20
DAFTAR PUSTAKA	21
LAMPIRAN	22
RIWAYAT HIDUP	48

DAFTAR TABEL

1 Data informasi harga komoditas pada aplikasi <i>mobile</i> Portal Harga	7
2 <i>User story</i> aplikasi <i>mobile</i> Portal Harga	9
3 Pengujian REST API dengan aplikasi <i>mobile</i> Portal Harga iterasi pertama	16
4 Pengujian REST API dengan aplikasi <i>mobile</i> Portal Harga iterasi kedua	20

DAFTAR GAMBAR

1 Skema konektivitas API antar <i>software</i> (3Scale Networks 2011)	3
2 <i>Roadmap</i> penelitian <i>e-Government</i> bidang pertanian	4
3 Tahapan dalam metode <i>Extreme Programming</i> (XP) (Pressman 2010)	6
4 Arsitektur perangkat lunak	8
5 <i>Use case diagram</i> aplikasi <i>mobile</i> Portal Harga	11
6 Detil <i>activity diagram</i> menambah informasi harga	11
7 <i>Class diagram</i> aplikasi <i>mobile</i> Portal Harga	12
8 Rancangan antarmuka halaman <i>login</i> aplikasi <i>mobile</i> Portal Harga	13
9 Rancangan antarmuka halaman utama masing-masing pengguna	13
10 Potongan <i>script</i> REST API untuk fungsi opini operasi pasar	14
11 Hasil keluaran <i>method</i> GET pada REST API dalam format JSON	15
12 Potongan <i>script</i> GET <i>parser</i> untuk menu <i>timeline</i> pada aplikasi	15
13 <i>Use case diagram</i> aplikasi <i>mobile</i> Portal Harga pada iterasi kedua	16
14 Rancangan antarmuka halaman utama pada iterasi kedua	17
15 Implementasi XML antarmuka halaman utama aplikasi Portal Harga	17
16 Potongan <i>script</i> REST API fungsi <i>post</i> aspirasi petani	18
17 Potongan <i>script</i> POST <i>parser</i> pada fungsi aspirasi petani	19
18 Hasil keluaran format JSON ketika <i>post</i> aspirasi tidak berhasil	19
19 Hasil keluaran format JSON ketika <i>post</i> aspirasi berhasil	19

DAFTAR LAMPIRAN

1 <i>Use case</i> deskripsi	22
2 <i>Activity diagram</i>	27
3 Rancangan antarmuka dan implementasi	31
4 Implementasi GET <i>parser</i>	40
5 Implementasi POST <i>parser</i>	43
6 Proses pengujian aplikasi Portal Harga	46

PENDAHULUAN

Latar Belakang

Sektor pertanian merupakan sektor yang berpengaruh dan mendominasi dalam perekonomian Indonesia. Hal ini dapat dilihat dari jumlah tenaga kerja di bidang pertanian yang mendominasi dengan jumlah mencapai 37.18 juta jiwa atau 32.61% dari jumlah tenaga kerja Indonesia seluruhnya pada Februari 2013 (Sekjen Kemenperin 2014). Menurut Undang-Undang RI Nomor 19 Tahun 2013 tentang Perlindungan dan Pemberdayaan Petani pada pasal 3 butir d menyatakan bahwa pemerintah wajib melindungi petani dari fluktuasi harga, praktik ekonomi biaya tinggi, dan gagal panen.

Permasalahan yang sering terjadi adalah fluktuasi harga yang disebabkan adanya pihak ketiga sebagai perantara, dalam hal ini adalah tengkulak serta keterbatasan pemerintah untuk mengetahui keadaan harga komoditas pertanian di pasar. Pada masa panen raya petani ingin menjual hasil panennya dengan cepat karena terdesak kebutuhan ekonomi atau modal untuk produksi selanjutnya. Akibat kurangnya informasi pemerintah mengenai status panen raya membuat pemerintah mengalami keterlambatan untuk membeli hasil panen sehingga petani terpaksa menjualnya kepada tengkulak agar terjual dengan cepat. Akibatnya harga yang diterima petani lebih rendah dari harga pemerintah sehingga petani dan masyarakat mengalami kerugian.

Menurut survey dari KPCB Internet Trend pada tahun 2015 pengguna *mobile phone* di Indonesia meningkat signifikan sebanyak 19% dari tahun sebelumnya. Indonesia sendiri menempati peringkat ke-9 untuk penggunaan *smartphone* terbanyak. Jumlah pengguna internet di Indonesia juga semakin berkembang dari tahun ke tahun. Survey APJII pada tahun 2014 mengungkapkan jumlah pengguna internet di Indonesia mencapai 88 juta orang dari 252.4 juta orang. Berdasarkan populasinya, jumlah pengguna internet terbanyak adalah di Pulau Jawa dan Bali dengan mayoritas pengguna adalah pekerja atau wiraswasta, mahasiswa, dan ibu rumah tangga (APJII 2014).

Jumlah pengguna *smartphone* dan internet yang besar merupakan potensi yang dapat dimanfaatkan untuk menanggulangi permasalahan tersebut. Salah satu pemanfaatan teknologi yang sudah dikembangkan adalah aplikasi *mobile* untuk memberikan data harga komoditas pertanian. Aplikasi tersebut bernama Pantau Harga yang dibuat oleh tim *Coding Stelsel* pada tahun 2015 (Code4Nation 2015). Pada aplikasi tersebut pengguna dapat memberikan data harga melalui *smartphone*, memberikan informasi harga komoditas dari penjual atau produsen kecil kepada konsumen untuk membeli barang, dan juga dapat melakukan pencarian harga yang terdekat dan termurah dari lokasi pengguna.

Penelitian mengenai sistem informasi pelaporan harga sebelumnya telah dilakukan, di antaranya oleh Astika *et al.* (2009) yaitu membangun sistem informasi harga produk pertanian berbasis *Short Message Service* (SMS). Sistem informasi ini menyediakan informasi harga produk pertanian bagi petani yang akan menjual hasil pertaniannya dan bagi pelaku agribisnis yang melakukan jual beli hasil pertanian melalui SMS. Penelitian yang dilakukan oleh Husen (2014)

yaitu mengenai sistem informasi pemetaan dan analisis distribusi harga produk pertanian di Indonesia berdasarkan tren komoditas per bulan dan tahunnya.

Penelitian ini mengembangkan modul pelaporan harga komoditas pertanian untuk petani, pemerintah, dan masyarakat. Pengembangan modul ini bertujuan untuk menyediakan media komunikasi berupa aspirasi petani terhadap pemerintah terkait pembelian hasil panen raya, membuat basis data informasi harga produk pertanian, serta memberikan laporan informasi berkala mengenai harga komoditas pertanian yang beredar di masyarakat. Pada pengembangan aplikasi ini menggunakan REST API karena lebih sederhana, tidak memerlukan struktur data khusus, dapat mempercepat aliran informasi, dan dapat berinteraksi dengan bahasa pemrograman lain (Reddy 2011).

Perumusan Masalah

Perumusan masalah dari penelitian ini adalah:

- 1 Bagaimana pengembangan modul pelaporan harga komoditas pertanian dapat menyediakan informasi harga dan lokasinya kepada masyarakat dan pemerintah?
- 2 Bagaimana membangun aplikasi pelaporan harga sehingga informasi harga komoditas pertanian dapat sampai dengan cepat kepada *stakeholder* terkait?
- 3 Bagaimana pemerintah mengetahui aspirasi petani mengenai hasil panen raya melalui aplikasi ini?

Tujuan Penelitian

Tujuan dari penelitian ini adalah:

- 1 Mengembangkan modul pelaporan harga komoditas pertanian menggunakan *Representational State Transfer* (REST API).
- 2 Mengembangkan aplikasi *mobile* pelaporan harga komoditas pertanian yang melibatkan interaksi pengguna masyarakat, petani, dan pemerintah.
- 3 Membangun aplikasi *mobile* yang dapat memberikan informasi mengenai aspirasi dan status panen petani kepada pemerintah.

Manfaat Penelitian

Manfaat dari penelitian ini adalah:

- 1 Dari sisi petani
 - a Memberikan informasi kepada pemerintah mengenai status produksi dan jumlah hasil panen.
 - b Memberikan aspirasinya terkait kebutuhan petani dalam memproduksi.
- 2 Dari sisi pemerintah
 - a Mengetahui aspirasi petani secara langsung dan *up-to-date*.
 - b Mengetahui harga yang beredar di pasar secara *real-time*.
- 3 Dari sisi masyarakat
 - a Memberikan informasi harga yang beredar di pasar secara langsung.
 - b Mengetahui informasi harga disekitarnya.

Ruang Lingkup Penelitian

Ruang lingkup dari penelitian ini adalah:

- 1 Aplikasi berbasis *mobile* untuk *platform* Android dengan spesifikasi minimal Jelly Bean 4.1.
- 2 Pengguna dari aplikasi Portal Harga ada tiga, yaitu petani komoditas beras, bawang, cabai, kedelai, dan daging sapi, pemerintah seperti Kementerian Pertanian yang bertanggung jawab dengan harga di pasar yaitu Dinas Perindustrian dan Perdagangan, serta masyarakat pengguna *smartphone platform* Android selaku konsumen.
- 3 Sistem tidak melakukan verifikasi data yang diberikan oleh pengguna. Lingkup pasar hanya di wilayah Jabodetabek, yaitu Pasar Induk Kramat Jati, Pasar Jatinegara, Pasar Sunter Podomoro, Pasar Grogol, Pasar Minggu, Pasar Senen, Pasar Kebon Kembang, Pasar Kemiri Muka, Pasar Induk Tanah Tinggi, dan Pasar Cikarang.
- 4 Mengimplementasikan REST API yang dibuat untuk masing-masing pengguna dan token API yang digunakan bersifat statis.

TINJAUAN PUSTAKA

Application Programming Interface (API)

API menyediakan abstraksi dari suatu masalah dan menspesifikasikan bagaimana *client* berinteraksi dengan komponen *software* yang mengimplementasikan solusi terhadap masalah tersebut. API merupakan *software interface* yang terdiri atas kumpulan instruksi yang disimpan dalam bentuk *library* dan menjelaskan bagaimana agar suatu *software* dapat berinteraksi dengan *software* lainnya (Reddy 2011).



Gambar 1 Skema konektivitas API antar *software* (3Scale Networks 2011)

Pada Gambar 1 dijelaskan fungsi API dalam menghubungkan antara *client* dan *server*. Penelitian ini menggunakan API Google Maps dan REST API web aplikasi Portal Harga.

Representational State Transfer (REST)

REST adalah gaya arsitektural yang memiliki aturan seperti antar muka yang seragam, sehingga jika aturan tersebut diterapkan pada *web services* akan dapat memaksimalkan kinerja *web services* terutama pada performa, skalabilitas, dan kemudahan untuk modifikasi (Kurniawan 2014). REST dapat digunakan sebagai *interface* dari API untuk mengakses suatu *resources*. REST menggunakan

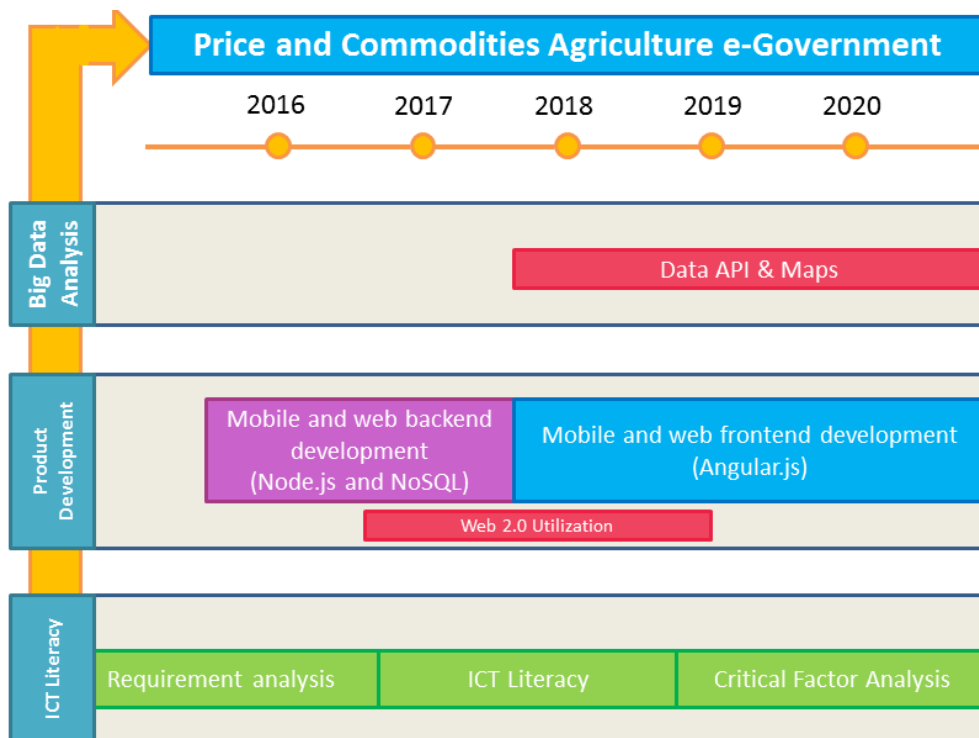
protokol *Hypertext Transfer Protocol* (HTTP) yang bersifat *stateless*. Perintah HTTP yang digunakan adalah fungsi GET, POST, PUT, dan DELETE.

Penggunaan REST API pada penelitian berfungsi sebagai jembatan data antara *client* dengan *server*. Hasil yang dikirimkan dari *server* berbentuk format *Java Script Object Notation* (JSON) tanpa ada protokol pemaketan data sehingga informasi yang diterima lebih mudah dibaca, ukuran *file* menjadi lebih kecil dan mudah diterima dari sisi *client* (Kurniawan 2014). JSON merupakan suatu format pertukaran data yang dirancang menjadi format yang mudah dibaca dan ditulis oleh manusia, serta mudah untuk diproses oleh komputer (Nurseitov *et al.* 2009).

Roadmap Penelitian

Roadmap adalah rencana kerja rinci yang menggambarkan seluruh rencana dan pelaksanaan program dalam jangka waktu tertentu. *Roadmap* digunakan dalam penelitian sebagai panduan untuk menentukan arah penelitian dan target yang akan dituju serta untuk monitoring pelaksanaan penelitian. Tim peneliti *e-Government* bidang pertanian pada Laboratorium *Software Engineering and Information Science* (SEInS) Departemen Ilmu Komputer FMIPA IPB memiliki *roadmap* untuk mengembangkan *e-Government* ini dalam rentang waktu tahun 2016 – 2020.

Roadmap penelitian ini memiliki tiga fokus kerja utama, yaitu *big data analysis*, *product development*, dan *ICT literacy*. Detil dari *roadmap* penelitian *e-Government* bidang pertanian dapat dilihat pada Gambar 2. Penelitian ini merupakan pelaksanaan dari *product development* bagian *mobile and web backend development* yang berfokus pada bagian *mobile* yang memiliki modul pelaporan harga komoditas pertanian yang melibatkan interaksi antar pengguna.



Gambar 2 *Roadmap* penelitian *e-Government* bidang pertanian

Undang-Undang RI Nomor 19 Tahun 2013

Dalam ketentuan umum Undang-Undang RI Nomor 19 Tahun 2013 tentang Perlindungan dan Pemberdayaan Petani menyatakan bahwa perlindungan petani adalah segala upaya untuk membantu petani dalam menghadapi permasalahan kesulitan memperoleh prasarana dan sarana produksi, kepastian usaha, risiko harga, kegagalan panen, praktik ekonomi biaya tinggi, dan perubahan iklim. Sedangkan pemberdayaan petani adalah segala upaya untuk meningkatkan kemampuan petani untuk melaksanakan usaha tani yang lebih baik melalui pendidikan dan pelatihan, penyuluhan dan pendampingan, pengembangan sistem dan sarana pemasaran hasil pertanian, konsolidasi dan jaminan luasan lahan pertanian, serta kemudahan akses ilmu pengetahuan, teknologi dan informasi.

Pada pasal 3 disebutkan salah satu tujuan dari perlindungan dan pemberdayaan petani dalam butir d untuk melindungi petani dari fluktuasi harga, praktik ekonomi biaya tinggi, dan gagal panen. Sesuai dengan pasal 7 nomor 2 butir c bahwa salah satu strategi perlindungan petani adalah harga komoditas pertanian dan pasal 13 bahwa pemerintah dan pemerintah daerah sesuai dengan kewenangannya bertanggung jawab atas perlindungan petani. Penelitian ini membuat aplikasi yang memiliki fungsi untuk memberikan informasi harga komoditas beserta lokasinya, serta informasi status panen dari petani. Sehingga dapat membantu pemerintah dalam melindungi petani terkait harga komoditas pertanian dan fluktuasi harga.

METODE PENELITIAN

Data Penelitian

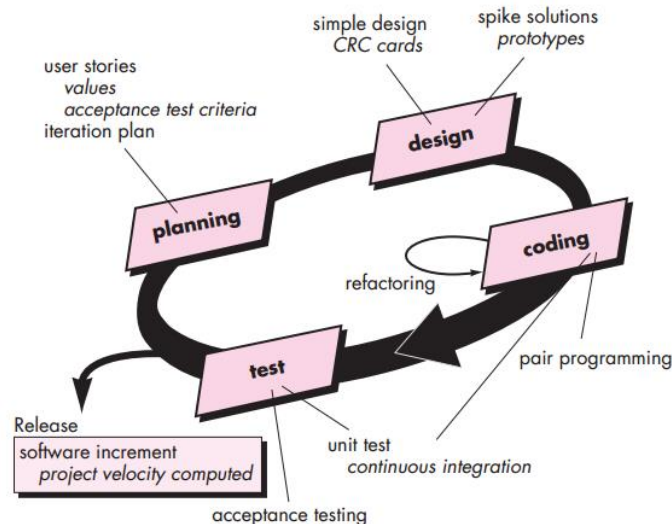
Data yang digunakan dalam penelitian ini terdiri dari data pelaporan harga komoditas pertanian yang berisi jenis komoditas, informasi lokasi, dan harga komoditas tersebut yang dimasukkan oleh masyarakat, serta data aspirasi dan pesan status produksi yang diperoleh dari masukkan petani. Pengguna aplikasi *mobile* Portal Harga dan data yang dimasukkan oleh pengguna merupakan data *dummy*.

Perancangan Arsitektur Penelitian

Perancangan arsitektur perangkat lunak yang digunakan pada penelitian terdapat tiga bagian, yaitu bagian data, *client*, dan *server*. Pada bagian data menggunakan basis data relasional dengan MySQL. Pada bagian *server* menggunakan Google Maps API sebagai penanda *latitude* dan *longitude* dari lokasi yang dimasukkan pengguna serta REST API untuk masing-masing pengguna. Pada bagian *client* terdapat dua keluaran, yaitu berbasis web dan *mobile*. Pada *client* yang berbasis web menggunakan bahasa pemrograman PHP Framework *Code Igniter* (CI) sebagai tampilan web, sedangkan yang berbasis *mobile* menggunakan bahasa pemrograman Java Android.

Tahapan Penelitian

Penelitian ini dilakukan dengan menggunakan metode *Extreme Programming* (XP) yang termasuk ke dalam *Agile Process Model*. Ciri dari *Agile Process Model* adalah tim yang tanggap terhadap perubahan karena perubahan adalah hal yang utama dalam pembangunan *software*, kebutuhan *software*, perubahan anggota tim, dan teknologi. Menurut Pressman (2010) tahapan pengembangan sistem dalam metode XP seperti pada Gambar 3. Tahapan yang dilakukan adalah:



Gambar 3 Tahapan dalam metode *Extreme Programming* (XP) (Pressman 2010)

1 Perencanaan (XP Planning)

Tahap perencanaan dimulai dari diskusi secara internal oleh tim untuk mendapatkan fitur atau fungsi utama yang akan dibuat. Hasil diskusi pada tahap ini berupa pembuatan *user story* dengan menentukan kebutuhan yang diprioritaskan untuk diselesaikan terlebih dahulu melalui indeks prioritas dan dibagi menjadi modul-modul yang lebih kecil. Pada tahap ini juga dilakukan analisis kebutuhan dasar sistem dengan cara studi literatur dan diskusi tim, serta menentukan banyaknya iterasi yang diperlukan dalam mengembangkan sistem.

2 Desain (XP Design)

Pada tahap ini dilakukan pembuatan *class diagram* untuk menggambarkan *class* dari objek yang ada didalam sistem, perancangan desain antarmuka pengguna dari *user story* yang disesuaikan dengan kebutuhan dan dekomposisi modul, serta implementasi desain XML aplikasi.

3 Pengkodean (XP Coding)

Tahapan ini membuat REST API dan pembuatan kode dari fungsi sistem yang akan digunakan. Pembuatan REST API dilakukan bersama oleh tim, yaitu Dian, Kadek, dan Shella. Pembuatan kode dari fungsi sistem meliputi implementasi *parser* REST API ke dalam android, serta pembuatan halaman aplikasi untuk setiap *class* objek.

4 Pengujian (XP Testing)

Setelah semua fungsi selesai dibangun dan digabungkan dengan aplikasi utama maka tahapan selanjutnya adalah dilakukan pengujian sistem. Pengujian sistem ini dilakukan dengan metode pengujian *black-box* dan *acceptance testing* yang dilakukan bersama secara internal oleh tim.

5 Release

Tahapan *release* sistem dilakukan setelah sistem sudah layak untuk dipublikasikan. Sistem dikatakan layak untuk dipublikasikan apabila sudah memenuhi kebutuhan pengujian sistem. Dalam tahapan *release* akan dilakukan implementasi sistem secara keseluruhan. Kode REST API diimplementasikan ke dalam *server* dan aplikasi android dibuat menjadi *Android Package* (APK) sehingga aplikasi dapat diakses dan digunakan oleh pengguna.

Lingkungan Pengembangan

Spesifikasi perangkat keras yang digunakan pada penelitian ini adalah laptop dengan spesifikasi *Processor* Intel® Core™ i5-3210M, RAM 8 GB, dan 1 TB HDD. *Smartphone* yang digunakan memiliki spesifikasi sistem operasi Android Jelly Bean 4.2.1, RAM 1 GB, dan *Processor* Quad Core 1.2 GHz MediaTek. Perangkat lunak yang digunakan adalah sistem operasi Windows 8 Pro 64 bit, Android Studio 1.4 sebagai *software* editor, Java Development Kit (JDK) 7, Nodejs 4.4.1, dan Visual Paradigm 12.2 Community Edition. Bahasa pemrograman yang digunakan adalah Java Android.

HASIL DAN PEMBAHASAN

Data Penelitian

Data yang digunakan dalam penelitian adalah data harga komoditas pertanian seperti beras, cabai, bawang, kedelai, dan daging sapi yang berada di pasar, pesan permintaan operasi pasar, dan opini petani untuk pemerintah. Pemilihan komoditas yang digunakan pada penelitian ini berdasarkan hasil observasi melalui internet dan Badan Pusat Statistik dengan kategori komoditas yang paling sering mengalami fluktuasi harga dan jumlah impor yang tinggi. Salah satu data yang digunakan adalah informasi harga komoditas yang diperoleh melalui REST API. Detil data harga komoditas dapat dilihat pada Tabel 1. Data harga komoditas terdiri dari delapan atribut, yaitu *id_komoditas*, nama, harga, tanggal, latitude, longitude, lokasi, dan *id_user*. Data tersebut diperoleh dari masukkan pengguna melalui aplikasi Portal Harga.

Tabel 1 Data informasi harga komoditas pada aplikasi *mobile* Portal Harga

Id	Nama	Harga	Tanggal	Latitude	Longitude	Lokasi	Id_user
1	Beras	8000	2016-05-27	-6.58604	106.72013	Dramaga	2

Arsitektur Penelitian

Arsitektur perangkat lunak yang digunakan dalam penelitian ini dapat dilihat pada Gambar 4. Pada arsitektur perangkat lunak ini terdapat tiga bagian, yaitu:

1 Data

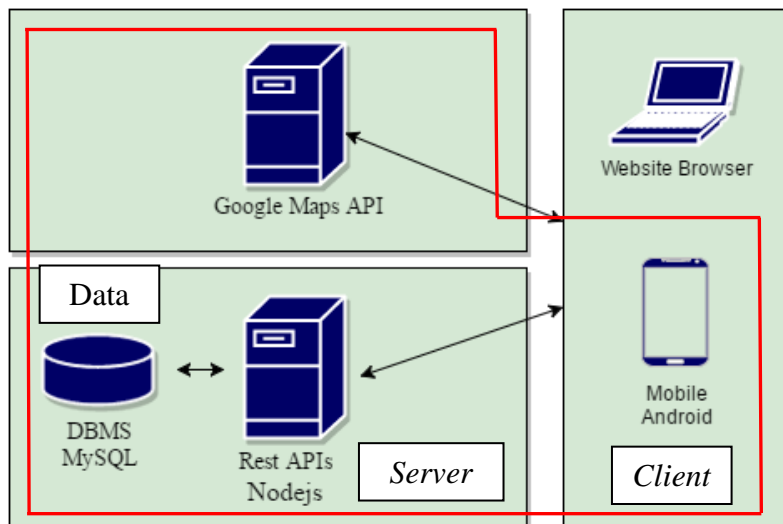
Pada bagian data, basis data yang digunakan adalah basis data relasional dengan MySQL. Basis data dikerjakan oleh tim yang membuat REST API, yaitu Dian Setiana, Kadek Grihadevi, dan Shellafuri Biru. Pada basis data terdapat tujuh tabel yang digunakan, yaitu tabel *user*, harga, lokasi, komoditas, pasar, opini, dan aspirasi.

2 Client

Pada bagian *client*, penelitian ini berfokus pada *mobile* untuk *platform* Android dengan menggunakan bahasa pemrograman Java Android. Bagian *client* dengan *website browser* dan pembuatan REST API dikerjakan oleh rekan satu tim, yaitu Dian Setiana, Kadek Grihadevi, dan Shellafuri Biru.

3 Server

Pada bagian *server*, terdapat basis data, Google Maps API, dan REST API. Google Maps API digunakan sebagai penanda *latitude* dan *longitude* dari suatu lokasi yang dimasukkan oleh pengguna. REST API digunakan sebagai jembatan antara basis data dengan aplikasi *mobile* menggunakan Node Js. Data yang disajikan pada aplikasi *mobile* Portal Harga berasal dari REST API pada *server*. Proses pengambilan data dari API dilakukan menggunakan fungsi GET dan POST. *Server* dalam penelitian ini menggunakan salah satu komputer yang ada di Departemen Ilmu Komputer FMIPA IPB.



Gambar 4 Arsitektur perangkat lunak

Tahapan Penelitian

Metode yang digunakan dalam pembuatan aplikasi pelaporan harga komoditas pertanian berbasis *mobile* adalah metode *Extreme Programming* (XP) yang termasuk ke dalam *Agile Process Model*. Metode ini terdiri dari empat tahap, yaitu tahap perencanaan, desain, pengkodean, dan pengujian. Pembuatan aplikasi pelaporan harga komoditas pertanian berbasis *mobile* mengalami dua kali iterasi *Extreme Programming* (XP).

1 Iterasi Pertama

1.1 Tahap Perencanaan (XP Planning)

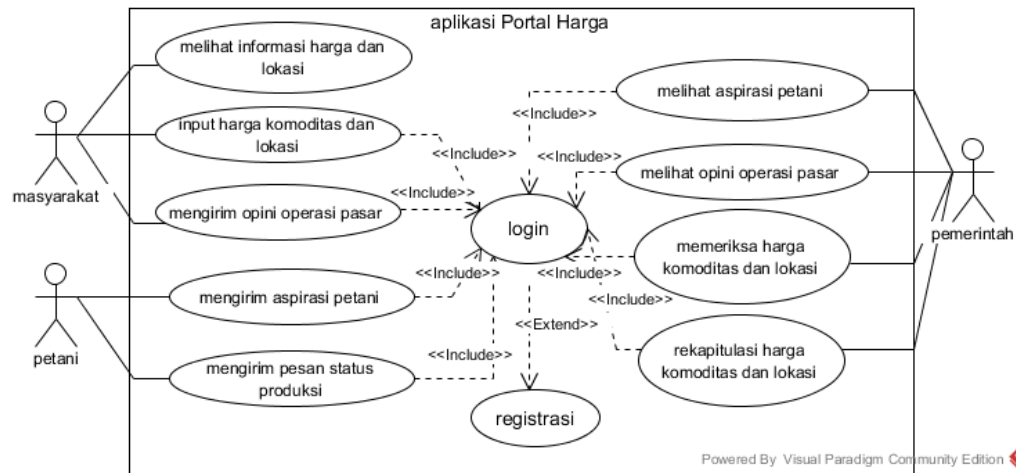
Awal mula tahap perencanaan dilakukan diskusi internal oleh tim, yaitu Dian, Kadek, dan Shella untuk menentukan kebutuhan dasar dari sistem yang akan dibuat. Hasil dari diskusi berupa pembuatan *user story* yang menggambarkan kebutuhan pengguna dengan memberikan prioritas dari kebutuhan tersebut. Hal ini bertujuan untuk memahami kebutuhan sistem dan memudahkan dalam pembuatan sistem. Detil dari *user story* aplikasi *mobile* Portal Harga dapat dilihat pada Tabel 2.

Tabel 2 *User story* aplikasi *mobile* Portal Harga

<i>Actor</i>	<i>Task</i>	<i>User Story</i>	Prioritas
Masyarakat	Menambah informasi harga komoditas	Pengguna memasukkan data harga dan lokasi dimana ia membeli komoditas, misalnya pengguna membeli bawang di pasar A, maka pengguna memasukkan data harga bawang di pasar A sehingga informasi mengenai harga bawang di pasar A dapat diketahui oleh pengguna lainnya.	1
	Melihat informasi harga komoditas	Masyarakat ingin mengetahui informasi harga komoditas di pasar dan lokasinya, sehingga bisa mendapatkan harga yang sesuai dengan kebutuhannya.	2
	Mengirim opini terkait operasi pasar	Terjadi lonjakan harga di pasar suatu daerah akibat kekurangan pasokan komoditas, maka masyarakat memberikan pesan kepada pemerintah untuk melakukan operasi pasar di daerah tersebut, sehingga lonjakan harga dapat diatasi lebih cepat.	3

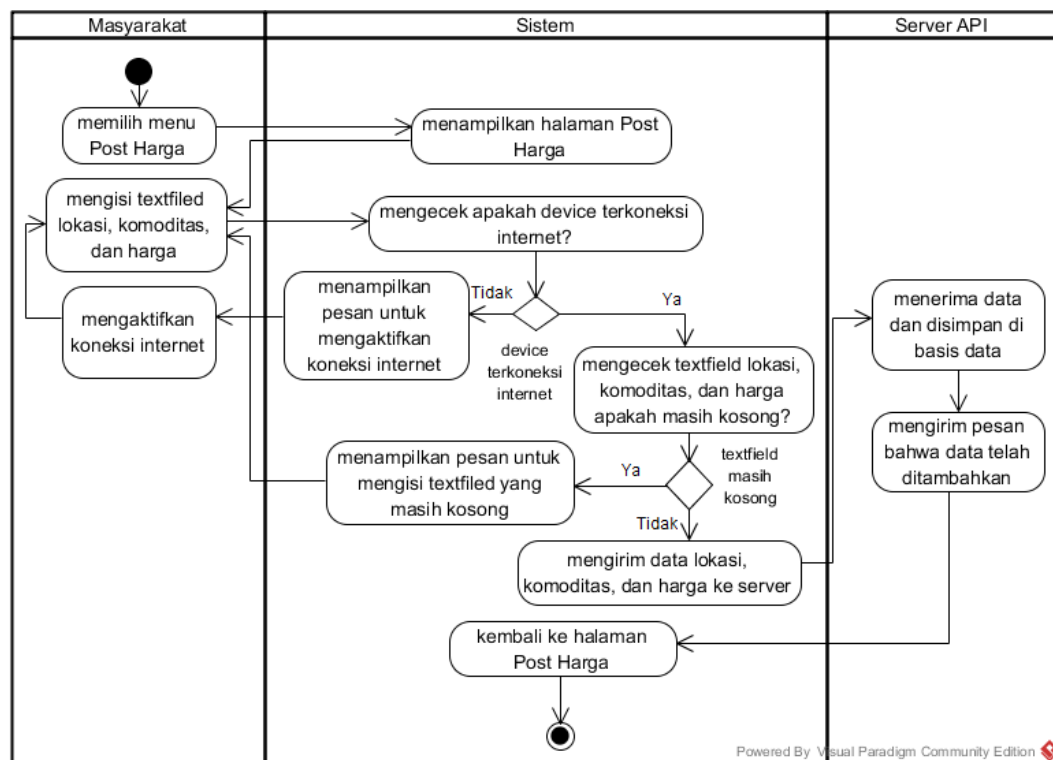
<i>Actor</i>	<i>Task</i>	<i>User Story</i>	<i>Prioritas</i>
Petani	Mengirim pesan status produksi	Ketika masa panen terjadi, petani ingin melaporkan status produksinya kepada pemerintah agar pemerintah bisa langsung membeli hasil panennya. Maka petani memberikan pesan status produksinya dengan memasukkan lokasi panen berada dan banyaknya hasil produksi.	1
	Mengirim aspirasi petani	Petani ingin menyampaikan aspirasi atau ide yang dimilikinya terkait peningkatan produksinya atau panen raya, maka petani memasukkan aspirasinya.	2
Pemerintah	Melihat opini terkait operasi pasar	Pemerintah akan mendapatkan notifikasi opini mengenai permintaan operasi pasar dari masyarakat. Notifikasi yang diterima berupa pesan permintaan serta lokasi operasi pasar seharusnya dilakukan.	1
	Melihat informasi harga komoditas dan lokasi	Pemerintah dapat memantau harga-harga komoditas yang berada di pasar dengan cara melihat informasi harga yang diinputkan oleh pengguna. Hal ini bertujuan untuk memudahkan pemerintah dalam mengambil kebijakan harga pasar.	2
	Melihat aspirasi dari petani dan masyarakat	Pemerintah mendapatkan aspirasi yang berupa saran atau kritikan dari petani dan masyarakat yang berguna untuk mengevaluasi harga serta komoditas yang berada di pasar.	3

User story berisi pengguna yang akan menggunakan aplikasi, yaitu masyarakat, petani, dan pemerintah, gambaran dari kebutuhan dari masing-masing pengguna, deskripsi penggunaan dari kebutuhan pengguna, serta prioritas kebutuhan pengguna. Dari hasil pembuatan *user story*, maka selanjutnya adalah membuat *use case diagram* aplikasi. *Use case diagram* yang dibuat untuk aplikasi *mobile* Portal Harga pada tahap ini mendefinisikan *stakeholder* yang terlibat dan kebutuhan fungsional dari masing-masing *stakeholder*. Detil dari *use case diagram* aplikasi *mobile* Portal Harga dapat dilihat pada Gambar 5. Dijelaskan dalam *use case diagram* terdapat tiga pengguna yang dapat menggunakan aplikasi *mobile* Portal Harga, yaitu masyarakat, petani, dan pemerintah.



Gambar 5 Use case diagram aplikasi mobile Portal Harga

Masyarakat memiliki tiga fungsi utama, yaitu melihat informasi harga, menambah informasi harga, dan mengirim opini operasi pasar. Petani memiliki dua fungsi utama, yaitu memasukkan aspirasi petani dan memberikan pesan status produksi panennya. Pemerintah memiliki empat fungsi utama, yaitu melihat informasi harga, melihat pesan opini, melihat pesan aspirasi, dan melihat rekapitulasi harga komoditas. Detil dari *use case diagram* dapat dilihat *use case* deskripsi pada Lampiran 1.



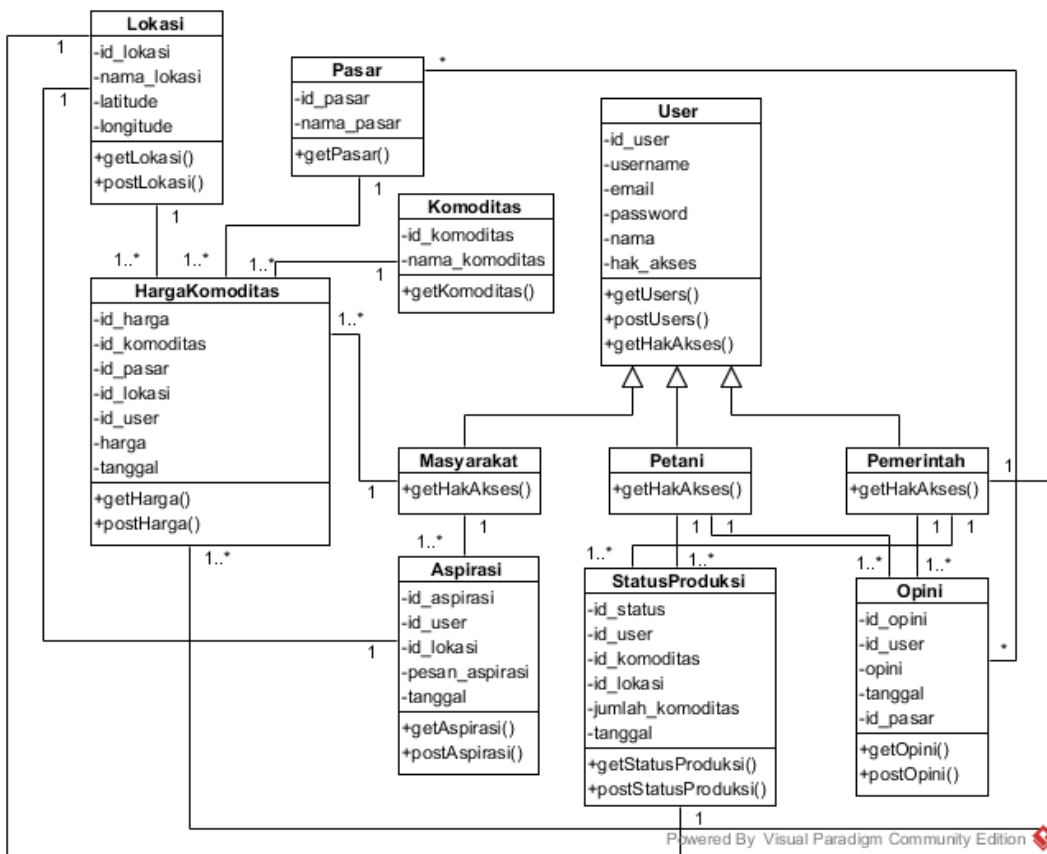
Gambar 6 Detil activity diagram menambah informasi harga

Pada *activity diagram* tersebut dijelaskan alur proses dari aktivitas menambah informasi harga melalui menu *post* harga. Data masukannya adalah lokasi, komoditas, dan harga komoditas yang dikirimkan melalui API dan masuk

ke dalam basis data di *server*. Detil dari *activity diagram* menambah informasi harga aplikasi *mobile* Portal Harga dapat dilihat pada Gambar 6. Rujukan *activity diagram* aplikasi *mobile* Portal Harga dapat dilihat pada Lampiran 2.

1.2 Tahap Desain (XP Design)

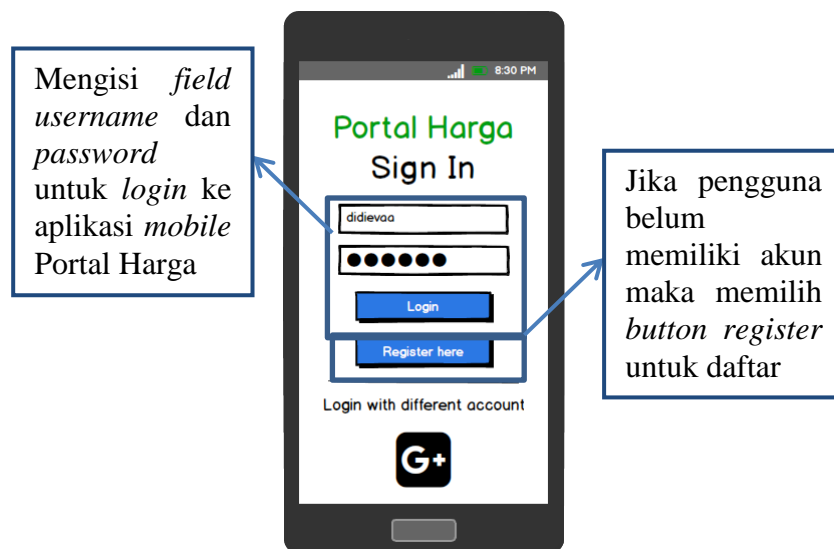
Tahap desain dilakukan berdasarkan skenario dan *use case diagram* yang telah dibuat pada tahap perencanaan. Pada tahapan ini mengerjakan perancangan *class diagram* sebagai *Class-Responsibility-Collaborator (CRC cards)* dan perancangan desain antarmuka sebagai *simple design*. *Class diagram* menggambarkan struktur dan deskripsi dari *class*, objek, *package*, serta hubungannya antar kelas. Pada aplikasi *mobile* Portal Harga terdapat delapan kelas utama, yaitu kelas *user*, harga komoditas, lokasi, pasar, opini, aspirasi, status produksi, dan komoditas. Pada kelas *user* terdapat penurunan kelas, yaitu kelas masyarakat, petani, dan pemerintah yang merupakan pengguna dari aplikasi. Detil dari *class diagram* dapat dilihat pada Gambar 7.



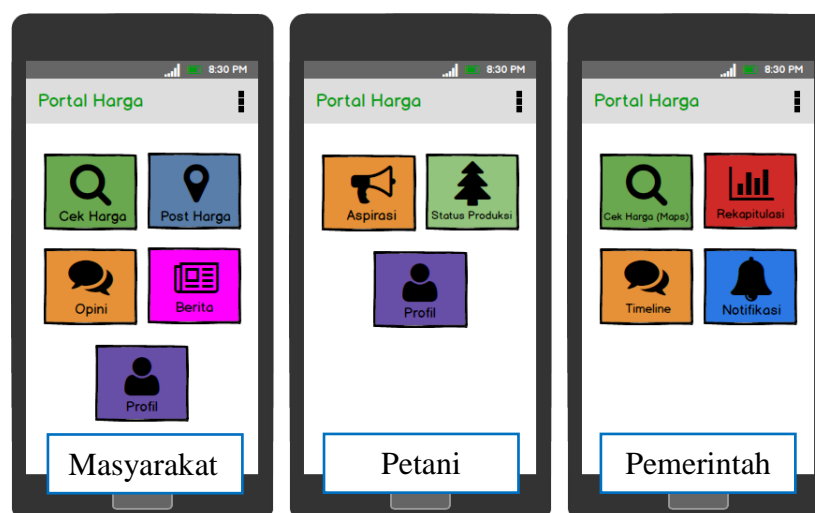
Gambar 7 Class diagram aplikasi *mobile* Portal Harga

Perancangan desain yang dibuat berupa *mock-up* atau rancangan antarmuka aplikasi yang menjadi acuan untuk tampilan antarmuka pengguna. Rancangan antarmuka halaman *login* dari aplikasi *mobile* Portal Harga dapat dilihat pada Gambar 8. Pada halaman *login*, pengguna memasukkan *username* dan *password* untuk masuk ke halaman utama. Jika pengguna belum mempunyai akun aplikasi maka pengguna harus melakukan registrasi terlebih dahulu.

Setelah pengguna berhasil *login*, maka pengguna akan masuk ke halaman utama. Terdapat tiga rancangan antarmuka halaman utama untuk masing-masing pengguna. Hal ini disesuaikan dengan kebutuhan masing-masing pengguna. Rancangan antarmuka halaman utama masing-masing pengguna aplikasi *mobile* Portal Harga dapat dilihat pada Gambar 9. Pada pengguna masyarakat terdapat lima menu, yaitu menu cek harga, *post* harga, opini, berita, dan profil. Pada pengguna petani terdapat tiga menu, yaitu menu aspirasi, status produksi, dan profil. Pada pengguna pemerintah terdapat empat menu, yaitu menu cek harga, *timeline*, notifikasi, dan rekapitulasi. Implementasi dari antarmuka aplikasi *mobile* Portal Harga sebagai pembuatan *spike solutions prototype*. Rujukan rancangan antarmuka aplikasi *mobile* Portal Harga dan implementasi desain XML antarmuka aplikasi dapat dilihat pada Lampiran 3.



Gambar 8 Rancangan antarmuka halaman *login* aplikasi *mobile* Portal Harga



Gambar 9 Rancangan antarmuka halaman utama masing-masing pengguna

1.3 Tahap Pengkodean (XP Coding)

Pada tahapan ini membuat fungsi utama dan REST API pada aplikasi Portal Harga. Salah satu contoh dari implementasi REST API fungsi Opini dapat dilihat pada Gambar 10. Implementasi REST API ini diakses melalui url `http://apps.cs.ipb.ac.id:4003/api_portalharga/request_operasi?token=portalharga` dengan *method* GET, API akan mengambil data dari basis data `portal_harga` yang menggabungkan data dari tabel *request*, *user*, *komoditas*, dan *pasar* dengan atribut yang ada pada masing-masing tabel. Jika *query* tidak berhasil dieksekusi maka API akan mengembalikan status error dan data tidak terambil. Jika *query* berhasil dieksekusi maka API akan mengambil data dari tabel yang dituju.

```

347 | router.get("/request_operasi", function(req, res) {
348 |     var query = "SELECT * from ?? join ?? on
        request.id_user=user.id_user join ?? on
        request.id_komoditas=komoditas.id_komoditas join ?? on
        request.id_pasar=pasar.id_pasar";
349 |     var table = ["request", "user", "komoditas", "pasar"];
350 |     query = mysql.format(query, table);
351 |     connection.query(query, function(err, rows) {
352 |         if (err) {
353 |             data.status = "error";
354 |             data.data = "error";
355 |         } else {
357 |             data.data = rows;
358 |         }
359 |         res.json(data);
360 |     });
361 | });

```

Gambar 10 Potongan *script* REST API untuk fungsi opini operasi pasar

Hasil keluaran dari *method* GET pada REST API dikirimkan oleh *server* dalam bentuk format JSON dapat dilihat pada Gambar 11. Format JSON tersebut terdiri dari JSON *array* dan JSON *object*. JSON *array* diawali dengan tanda kurung siku (“[]”), sedangkan pada JSON *object* diawali dengan tanda kurawal (“{ }”). JSON *array* adalah data dan di dalam data terdapat JSON *object*, yaitu `id_request`, `subjek`, `tanggal`, `pesan`, sampai dengan `nama_pasar`.

`Id_request` adalah id dari pesan *request* yang masuk ke dalam basis data. `Subjek` adalah masukkan dari pengguna mengenai judul pesan *request*. `Tanggal` adalah tanggal pesan *request* dimasukkan oleh pengguna. `Pesan` adalah pesan *request* yang dimasukkan oleh pengguna. `Id_user` adalah id dari pengguna yang melakukan *request*. `Id_pasar` dan `nama pasar` adalah pasar yang ingin dilakukan operasi pasar oleh pengguna. `Username`, `password`, `nama`, `email`, dan `hak_akses` adalah data diri dari pengguna.

Pada tahapan ini juga dibuat fungsi GET *parser* dari REST API yang telah dibuat untuk ditampilkan ke dalam aplikasi. Implementasi *parser* digunakan untuk menerjemahkan bahasa pemrograman JSON ke dalam Java menggunakan *library* Volley sehingga dapat digunakan untuk mengambil dan mengirimkan data ke basis data. Salah satu contoh GET *parser* dapat dilihat pada Gambar 12.


```

1  "data": [
2    {
3      "id_request": 6,
4      "subjek": "Bawang merah mahal",
5      "tanggal": "2016-05-25T17:00:00.000Z",
6      "pesan": "Pak bawang merah di pasar sunter podomoro naik
       banget. ",
7      "id_user": 4,
8      "id_pasar": 3,
9      "username": "kadek",
10     "password": "9bd0a034baadfadd3a89af39bc4320b",
11     "nama": "kadek",
12     "email": "kadek@gmail.com",
13     "hak_akses": 0,
14     "nama_pasar": "Pasar Sunter Podomoro"
15   }]

```

Gambar 11 Hasil keluaran *method* GET pada REST API dalam format JSON

Pada GET *parser* menggunakan JSONArray dan JSONObject untuk memisahkan JSON *array* dan JSON *object* sehingga data lebih mudah untuk dibaca. Jika JSON string tidak bernilai null maka mengambil JSON *array* data, kemudian dilakukan proses *looping* untuk mengambil JSON *object* yang ada di dalam JSON *array*. JSON *object* yang diambil adalah nama, jenis, nama_pasar, dan pesan yang dimasukkan ke dalam array HashMap bernama timelineList. Detil implementasi GET *parser* pada aplikasi *mobile* Portal Harga dapat dilihat pada Lampiran 4.

Pada tahapan ini dilakukan *pair programming* dimana pembuatan koding dilakukan secara bersama-sama. Dalam hal ini terdapat dua tim, yaitu tim pertanian dan kampanye dengan arsitektur penelitian yang sama. *Pair programming* dilakukan bersama dengan tim kampanye yang berbasis *mobile* untuk mengimplementasikan GET *parser* untuk masing-masing fungsi aplikasi.

```

101 String jsonStr = sh.makeServiceCall(url, ServiceHandler.GET);
102 Log.d("Response: ", "> " + jsonStr);
103 if (jsonStr != null) {
104   try {
105     JSONObject jsonObj = new JSONObject(jsonStr);
106     // Getting JSON Array data
107     data = jsonObj.getJSONArray(TAG_DATA);
108     // looping through All Data
109     for (int i = 0; i < data.length(); i++) {
110       JSONObject t = data.getJSONObject(i);
111       String nama = t.getString(TAG_NAMA);
112       String jenis = t.getString(TAG_JENIS);
113       String nama_pasar = t.getString(TAG_NAMA_PASAR);
114       String pesan = t.getString(TAG_PESAN);
115       // adding data timeline to timeline list
116       timelineList.add(data_timeline);
117     }

```

Gambar 12 Potongan *script* GET *parser* untuk menu *timeline* pada aplikasi

1.4 Tahap Pengujian (XP Testing)

Tahap pengujian yang dilakukan dalam pembuatan aplikasi Portal Harga ini menggunakan metode pengujian *black-box* yang dilakukan secara internal oleh

tim. Dalam hal ini dilakukan *unit test* dengan tim yang berbasis web, yaitu Dian, Kadek, dan Shella. *Unit test* yang dilakukan adalah melakukan pengujian pada masing-masing fungsi sistem untuk menyamakan hasil yang diterima oleh sistem. Pada tahap pengujian ini terdapat fungsi yang belum berhasil, yaitu fungsi Cek Harga dengan *Google Maps*. Hal ini dikarenakan *Google Maps API Key* yang belum terdaftar dan belum dapat diaktifkan pada aplikasi sehingga *Google Maps* belum bisa ditampilkan dalam halaman cek harga. Detil pengujian REST API dengan aplikasi *mobile* Portal Harga dapat dilihat pada Tabel 3. Pada tahapan *release* sistem dilakukan implementasi REST API ke dalam *server* dan menggunakan port 4003 dari domain *apps.cs.ipb.ac.id*. *Release* aplikasi *mobile* Portal Harga pada iterasi pertama belum dilakukan karena masih terdapat fungsi POST yang belum diimplementasikan dalam aplikasi.

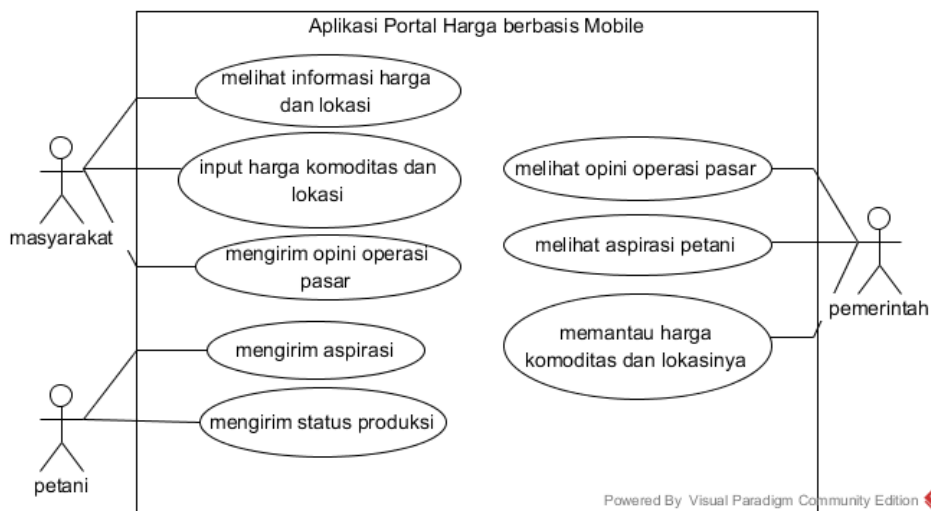
Tabel 3 Pengujian REST API dengan aplikasi *mobile* Portal Harga iterasi pertama

Actor	Fungsi REST	Status Pengujian
Masyarakat	GET Cek Harga	Belum berhasil
Pemerintah	GET Informasi Harga	Berhasil
	GET Informasi Harga (List)	Berhasil
	GET Notifikasi Status Produksi	Berhasil
	GET <i>Timeline</i> Operasi Pasar	Berhasil

2 Iterasi Kedua

2.1 Tahap Perencanaan (XP Planning)

Pada tahap perencanaan di iterasi kedua, dilakukan perencanaan ulang terhadap fungsi cek harga dengan *Google Maps* yang belum berhasil serta implementasi POST *parser* yang belum dibuat pada iterasi pertama. Terdapat perubahan fungsi pada sisi masyarakat, yaitu menghapus menu berita dikarenakan sulitnya mendapatkan berita mengenai pertanian yang selalu *update*. Pada *use case diagram* juga dilakukan perubahan, dimana semua pengguna harus *login* untuk masuk ke aplikasi Portal Harga. Perubahan pada *use case diagram* aplikasi *mobile* Portal Harga dapat dilihat pada Gambar 13.

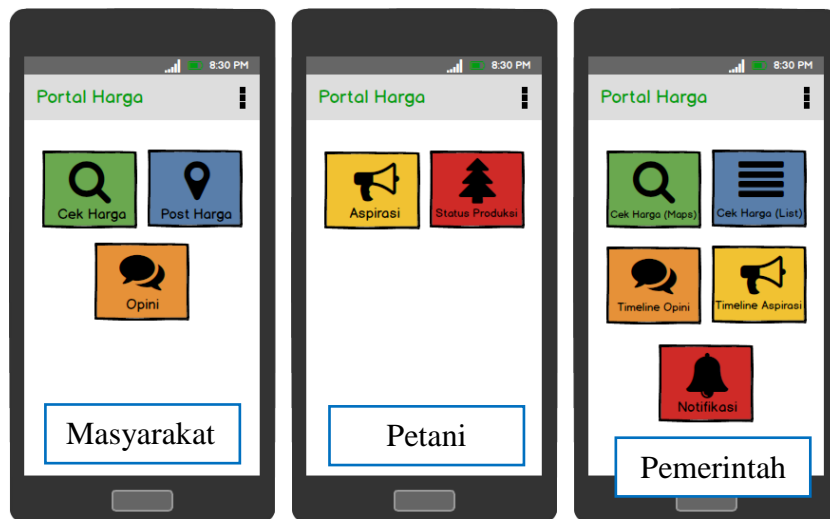


Gambar 13 *Use case diagram* aplikasi *mobile* Portal Harga pada iterasi kedua

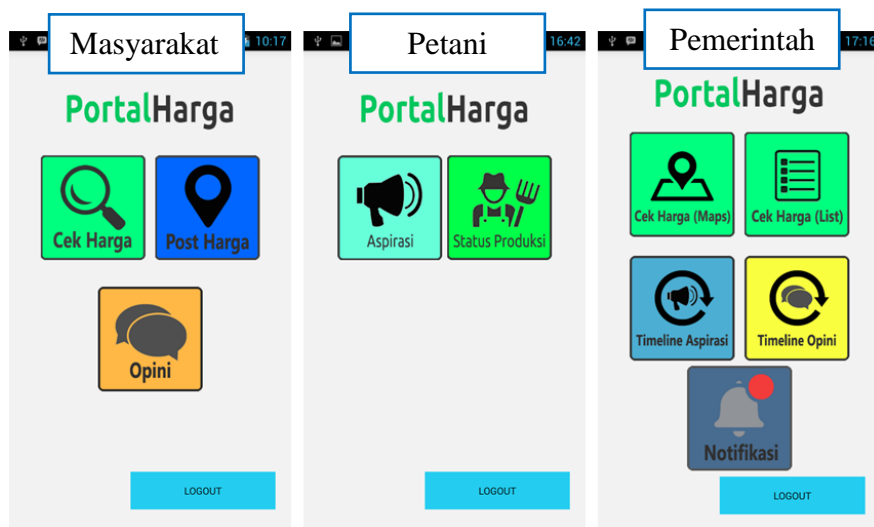
2.2 Tahap Desain (XP Design)

Pada tahap desain dilakukan pembuatan rancangan antarmuka aplikasi yang disesuaikan dengan tahap perencanaan sebelumnya. Pada tahapan ini tidak mengalami banyak perubahan antarmuka aplikasi dari sebelumnya, yaitu menghapus menu berita, profil, dan perbaikan beberapa tampilan pada halaman utama. Perubahan rancangan antarmuka halaman utama pengguna dapat dilihat pada Gambar 14. Implementasi dalam bentuk XML dari rancangan antarmuka halaman utama dapat dilihat pada Gambar 15.

Masyarakat memiliki tiga menu, yaitu menu cek harga, menu *post* harga, dan menu opini. Menu cek harga adalah menu yang berisi lokasi informasi harga pada Google Maps. Menu *post* harga adalah menu yang digunakan untuk pengguna mengirim informasi harga komoditas dan lokasinya. Menu opini adalah menu untuk mengirim pesan *request* operasi pasar. Petani memiliki dua menu, yaitu menu aspirasi dan menu status produksi. Menu aspirasi adalah menu untuk mengirim aspirasi petani, sedangkan menu status produksi adalah menu untuk memberitahukan mengenai status produksi dari hasil panen petani.



Gambar 14 Rancangan antarmuka halaman utama pada iterasi kedua



Gambar 15 Implementasi XML antarmuka halaman utama aplikasi Portal Harga

Pemerintah memiliki lima menu, yaitu menu cek harga dalam *maps* dan *list*, menu *timeline* aspirasi dan opini, serta menu notifikasi. Pada menu cek harga pemerintah sama seperti menu pada masyarakat, berupa *maps* dan daftar. Pada menu *timeline* aspirasi dan opini, pemerintah dapat melihat aspirasi dan opini yang dimasukkan oleh masyarakat dan petani. Menu notifikasi adalah menu untuk melihat pesan status produksi dari hasil panen raya petani.

2.3 Tahap Pengkodean (XP Coding)

Pada tahap pengkodean membuat fungsi yang belum berjalan pada iterasi pertama, yaitu GET cek harga dan *method* POST. Selain itu pada tahap ini dilakukan penambahan token OAuth pada REST API yang berfungsi untuk keamanan data API. Token yang digunakan dalam REST API berupa token statis sehingga token yang diakses selalu bernilai sama. Salah satu contoh struktur url untuk mengakses API adalah http://apps.cs.ipb.ac.id:4003/api_portalharga/post_aspirasi?token=portalharga dengan *method* POST. Salah satu contoh implementasi REST API fungsi *post* aspirasi petani dapat dilihat pada Gambar 16.

```

259 | router.post("/post_aspirasi", function(req,res){
260 |   var query="INSERT INTO ??(??,??,??,??) VALUES (?, ?, ?, ?)";
261 |   var table = ["aspirasi_petani", "id_user", "subject",
                "aspirasi_pesan", "time", req.body.id_user,
                req.body.subject, req.body.aspirasi_pesan, req.body.time];
262 |   query = mysql.format(query,table);
263 |   connection.query(query,function(err,rows){
264 |     if(err){
265 |       res.json({"Error" : true,"Message": "Error executing
                MySQL query"});
266 |     }else{
267 |       res.json({"Error" : false, "Message" : "Aspirasi
                Adedd !"});
268 |     }
269 |   });

```

Gambar 16 Potongan *script* REST API fungsi *post* aspirasi petani

Implementasi REST API ini diakses melalui url `/post_aspirasi` dengan *method* POST, API akan menambahkan data yang dimasukkan oleh pengguna ke dalam tabel `aspirasi_petani` dengan atribut `id_user`, `subject`, dan `aspirasi_pesan`. Implementasi POST *parser* untuk fungsi *post* aspirasi petani melalui fungsi `kirimAspirasi()` dapat dilihat pada Gambar 17. Pada POST *parser* menggunakan `HashMap` berupa *array* string sehingga setiap data yang dimasukkan akan dijadikan JSON Array.

Fungsi `kirimAspirasi()` mengirimkan `id_user`, `subject`, dan `aspirasi_pesan` yang dimasukkan oleh pengguna. Data yang dimasukkan dijadikan string dan disimpan ke dalam `HashMap array`. Setelah disimpan di dalam *array* maka data akan dijadikan JSON *array*. Jika data yang dimasukkan tidak berhasil maka akan muncul status eror berupa format JSON seperti pada Gambar 18. Jika data berhasil dimasukkan maka akan muncul pesan berhasil berupa format JSON seperti pada Gambar 19.

Pada tahapan ini dilakukan *pair programming* dimana pembuatan koding dilakukan bersama secara internal dengan tim yang berbasis *mobile*. Dalam hal ini

dilakukan pembuatan *POST parser* untuk masing-masing fungsi. Detil implementasi *POST parser* pada aplikasi mobile Portal Harga dapat dilihat pada Lampiran 5.

```

70 | private void kirimAspirasi(){
71 |     final String id_user = txtNama.getText().toString();
72 |     final String subject = txtSubjek.getText().toString();
73 |     final String aspirasi_pesan =
       txtAspirasi.getText().toString();
74 |
75 |     StringRequest stringRequest = new
       StringRequest(Request.Method.POST, URL_ASPIRASI,
76 |         new Response.Listener<String>() {
77 |             @Override
78 |             public void onResponse(String response) {
79 |                 Toast.makeText(MenuAspirasi.this, response,
                   Toast.LENGTH_LONG).show();
80 |             }
81 |         },
82 |         new Response.ErrorListener() {
83 |             @Override
84 |             public void onErrorResponse(VolleyError error) {
85 |                 Toast.makeText(MenuAspirasi.this, error.toString(),
                   Toast.LENGTH_LONG).show();
86 |             }
87 |         }) {
88 |         @Override
89 |         protected Map<String,String> getParams(){
90 |             Map<String,String> params = new HashMap<String, String>();
91 |             params.put(KEY_ID_USER,id_user);
92 |             params.put(KEY_SUBJECT, subject);
93 |             params.put(KEY_ASPIRASI_PESAN, aspirasi_pesan);
94 |             return params;
95 |         }
97 |     };

```

Gambar 17 Potongan *script POST parser* pada fungsi aspirasi petani

```

1 | {
2 |     "Error": true,
3 |     "Message": "Error executing MySQL query"
4 | }

```

Gambar 18 Hasil keluaran format JSON ketika *post* aspirasi tidak berhasil

```

1 | {
2 |     "Message": "Data berhasil ditambahkan, terima kasih!"
3 | }

```

Gambar 19 Hasil keluaran format JSON ketika *post* aspirasi berhasil

2.4 Tahap Pengujian (XP Testing)

Pengujian yang dilakukan pada iterasi kedua menggunakan pengujian *black box*, seperti iterasi pertama. Pada tahapan ini semua fungsi yang telah dibuat diuji secara internal oleh tim dan semua fungsi dapat berjalan dengan baik. Hasil pengujian dapat dilihat pada Tabel 4. Proses pengujian secara lengkap dapat

dilihat pada Lampiran 6. Pada tahap *release* sistem dilakukan implementasi aplikasi berupa pembuatan *Android Package* (APK) yang dapat digunakan di berbagai *smartphone*.

Tabel 4 Pengujian REST API dengan aplikasi *mobile* Portal Harga iterasi kedua

<i>Actor</i>	Fungsi REST	Status Pengujian
Masyarakat	GET Cek Harga	Berhasil
	POST Post Harga	Berhasil
	POST Opini operasi pasar	Berhasil
Petani	POST Aspirasi Petani	Berhasil
	POST Status Produksi Petani	Berhasil
Pemerintah	GET Informasi Harga	Berhasil
	GET Informasi Harga (List)	Berhasil
	GET Notifikasi Status Produksi	Berhasil
	GET <i>Timeline</i> Operasi Pasar	Berhasil

Release sistem dilakukan dengan beberapa tahap, yaitu dilakukan konfigurasi akhir dengan menggabungkan modul setiap *user* ke dalam satu aplikasi. Setelah itu, dilakukan uji coba dengan memasukkan data sesuai dengan ketentuan dan dicek melalui REST API. Setelah semua uji coba berhasil maka aplikasi diubah menjadi APK sehingga aplikasi bisa di instalasi di berbagai *smartphone*.

SIMPULAN DAN SARAN

Simpulan

Aplikasi Portal Harga berbasis *mobile* dibuat dengan menggunakan metode *Extreme Programming* (XP) sebanyak dua iterasi. Aplikasi *mobile* Portal Harga berhasil mengimplementasikan REST API ke dalam fungsi-fungsi yang tersedia, salah satunya adalah memberikan informasi mengenai harga komoditas beserta lokasinya dengan Google Maps. Fungsi dari REST API pada aplikasi *mobile* Portal Harga adalah dapat mengalirkan informasi lebih cepat tanpa harus mengambil data dari basis data terlebih dahulu. Aplikasi ini melibatkan interaksi dari ketiga pengguna, yaitu masyarakat, petani, dan pemerintah. Setiap pengguna aplikasi *mobile* Portal Harga memiliki fungsinya tersendiri. Masyarakat dapat memberikan informasi harga komoditas dan lokasinya, serta opini operasi pasar. Petani dapat mengirimkan aspirasi dan status produksi panennya. Pemerintah dapat melihat informasi-informasi yang dikirimkan oleh masyarakat dan petani.

Saran

Aplikasi *mobile* Portal Harga dari sisi pemerintah sebaiknya dapat memberikan tanggapan mengenai aspirasi dan opini dari masyarakat dan petani sehingga lebih interaktif. Aplikasi ini memerlukan pengembangan untuk tampilan antarmuka, salah satunya dengan menggunakan AngularJS agar lebih mudah diimplementasikan dan lebih menarik dari sisi pengguna. Selain itu perlu untuk

menambahkan keamanan data dengan cara membuat token OAuth menjadi token yang dinamis, serta melakukan integrasi *login* dengan *social media* sehingga dapat membantu memvalidasi informasi yang diberikan sehingga pengguna memperoleh informasi yang benar.

DAFTAR PUSTAKA

- 3Scale Networks. 2011. What is an API?. [Internet]. [diunduh 2016 Januari 17] . Tersedia pada: <http://www.3scale.net/wp-content/uploads/2012/06/What-is-an-API-1.0.pdf>
- [APJII] Asosiasi Penyelenggara Jasa Internet Indonesia. 2014. Profil Pengguna Internet Indonesia 2014. [Internet]. [diunduh 2015 November 18]. Tersedia pada: <http://www.slideshare.net/internetsehat/profil-pengguna-internet-indonesia-2014-riset-oleh-apjii-dan-puskakom-ui>
- Astika IW, Solahudin M, Maradona RF, Nugraha MPA. 2009. Sistem informasi harga produk pertanian berbasis SMS. Di dalam: *Seminar Nasional Himpunan Informatika Pertanian Indonesia 2009*; 2009 Agustus 6-7; Bogor (ID): Departemen Pertanian Republik Indonesia. hlm 1-7
- Code4Nation. 2015. Aplikasi Pantau Harga. [Internet]. [diunduh pada 2015 November 15]. Tersedia pada: <http://www.slideshare.net/ivanwidodo/aplikasi-pantau-harga>
- Husen. 2014. Pemetaan dan analisis distribusi harga produk-produk pertanian menggunakan sistem informasi geografis [skripsi]. Bogor (ID): Institut Pertanian Bogor.
- [KPCB] Kleiner Perkins Caufield Byers. 2015. Internet Trends Report. [internet]. [diunduh pada 2016 Januari 16]. Tersedia pada: <http://www.kpcb.com/internet-trends>
- Kurniawan E. 2014. Implementasi rest web service untuk sales order dan sales tracking berbasis mobile. *Jurnal Eksis*. 07(1):1-12.
- Nurseitov N, Paulson M, Reynolds R, Izurieta C. 2009. Comparison of JSON and XML data interchange formats: A case study. *Caine*. 9:157-162.
- Pressman RS. 2010. *Software Engineering: A Practitioner's Approach*. Ed ke-7. New York (US): McGraw-Hill
- Reddy M. 2011. *API Design for C++*. Burlington (US): Elsevier
- [Sekjen Kemenperin] Sekretariat Jenderal Kementerian Pertanian. 2013. *Statistik Ketenagakerjaan Sektor Pertanian Tahun 2013*. Jakarta (ID): Pusat Data dan Sistem Informasi Pertanian
- Undang-Undang Republik Indonesia Nomor 19 tahun 2013 Tentang Perlindungan dan Pemberdayaan Petani. [Internet]. [diunduh pada 2015 Desember 06]. Tersedia pada: http://www.dpr.go.id/dokjdih/document/uu/UU_2013_19.pdf

Lampiran 1 *Use case* deskripsia *Login*

No	<i>Use case</i>	Deskripsi
1	Nama	<i>Login</i>
	Deskripsi	Proses untuk memverifikasi bahwa pengguna yang akan menggunakan aplikasi sudah terdaftar dalam basis data atau belum
	Prekondisi	Halaman utama aplikasi tidak bisa diakses sebelum pengguna login terlebih dahulu
	Proses	Pengguna memasukkan <i>username</i> dan <i>password</i> kemudian menekan tombol <i>login</i> . Setelah itu aplikasi akan mengirimkan <i>username</i> dan <i>password</i> ke <i>server</i> API untuk dicek. Jika <i>username</i> dan <i>password</i> sudah terdaftar maka proses <i>login</i> berhenti sampai disini. Jika belum terdaftar maka pengguna akan kembali ke halaman <i>login</i> untuk mendaftarkan diri terlebih dahulu.
	Kondisi Akhir	Setelah proses <i>login</i> berhasil maka aplikasi akan menampilkan halaman utama Portal Harga

b *Register*

No	<i>Use case</i>	Deskripsi
2	Nama	<i>Register</i>
	Deskripsi	Pengguna belum mempunyai akun untuk masuk ke dalam aplikasi. Maka pengguna mendaftarkan dirinya untuk membuat akun baru.
	Prekondisi	Halaman utama dari aplikasi tidak bisa diakses dan mewajibkan pengguna untuk <i>login</i> . Jika pengguna belum mempunyai akun maka pengguna harus melakukan registrasi akun baru pada halaman Register.
	Proses	Pengguna memilih halaman Register dan memasukkan <i>username</i> , email, dan <i>password</i> . Apabila <i>username</i> dan email sudah terdaftar maka akan menampilkan pesan bahwa <i>username</i> dan email yang dimasukkan sudah digunakan. Jika belum terdaftar maka <i>username</i> , email, dan <i>password</i> akan disimpan ke dalam basis data melalui API.
	Kondisi Akhir	Setelah proses <i>register</i> berhasil, maka aplikasi akan menampilkan halaman <i>login</i> .

Lampiran 1 *Use case* deskripsi (lanjutan)

c Melihat informasi harga

No	<i>Use case</i>	Deskripsi
3	Nama	Melihat informasi harga
	Deskripsi	Pengguna melihat informasi harga komoditas melalui menu Cek Harga pada halaman utama. Pengguna dapat melihat informasi dimana lokasi penjual komoditas berada, jenis komoditas, harga jual komoditas tersebut, serta tanggal komoditas tersebut ditambahkan.
	Prekondisi	Pengguna berada di halaman utama dan memilih menu Cek Harga. Dalam hal ini, <i>device</i> harus dalam kondisi terkoneksi ke internet dan GPS menyala atau aktif.
	Proses	Pengguna melihat informasi harga komoditas yang ada di menu Cek Harga. Pengguna melihat lokasi informasi harga komoditas yang ada di sekitarnya, jenis komoditas yang ada, harga komoditas, serta tanggal harga komoditas tersebut berlaku
	Kondisi Akhir	Aplikasi akan menampilkan informasi lokasi, harga komoditas, jenis komoditas, dan tanggal harga komoditas tersebut di <i>update</i> .

d Menambah informasi harga

No	<i>Use case</i>	Deskripsi
4	Nama	Tambah informasi harga
	Deskripsi	Menambahkan informasi harga komoditas pertanian melalui menu <i>Post Harga</i> di halaman utama. Pengguna memberikan informasi lokasi penjual komoditas, harga komoditas tersebut, serta jenis komoditas tersebut.
	Prekondisi	Pengguna berada pada halaman utama dan memilih menu Post Harga. Dalam hal ini, <i>device</i> harus dalam kondisi terkoneksi ke internet dan GPS menyala atau aktif.
	Proses	Pengguna memilih menu <i>Post Harga</i> pada halaman utama. Kemudian pengguna memberikan informasi seperti lokasi dimana komoditas dijual, harga komoditas yang dijual, jenis komoditas yang dijual, serta tanggal pengguna menambahkan informasi harga komoditas tersebut. Aplikasi akan mengambil data titik <i>latitude</i> dan <i>longitude</i> lokasi komoditas ditambahkan.
	Kondisi Akhir	Aplikasi akan menambahkan informasi mengenai komoditas yang ditambahkan oleh pengguna

Lampiran 1 *Use case* deskripsi (lanjutan)

e Mengirim opini operasi pasar

No	<i>Use case</i>	Deskripsi
5	Nama	Mengirim opini operasi pasar
	Deskripsi	Mengirimkan pesan berupa opini atau aduan operasi pasar dari masyarakat untuk disampaikan kepada pemerintah.
	Prekondisi	Pengguna memilih menu Opini dari halaman utama. Jika pengguna ingin mengirimkan pesan baru maka pengguna harus memasukkan nama pasar yang akan dilakukan operasi pasar, subjek pesan, serta pesan operasi pasar.
	Proses	Aplikasi akan mengirimkan pesan atau opini dari pengguna ke pemerintah.
	Kondisi Akhir	Pesan masuk ke basis data melalui API dan dapat dilihat di menu <i>Timeline</i> Opini akun pemerintah.

f Mengirim pesan aspirasi

No	<i>Use case</i>	Deskripsi
6	Nama	Mengirim aspirasi petani
	Deskripsi	Pengguna dapat mengirimkan pesan aspirasinya kepada pemerintah mengenai kritik, saran, dan keluhannya.
	Prekondisi	Pengguna berada di halaman utama dan memilih menu Aspirasi. Jika pengguna ingin mengirimkan pesan aspirasinya maka pengguna harus memasukkan nama, subjek, dan pesan aspirasinya.
	Proses	Pengguna mengirimkan pesan aspirasinya dengan mengisi nama, subjek, dan pesan aspirasinya. Kemudian pengguna menekan tombol kirim.
	Kondisi Akhir	Aplikasi akan mengirimkan pesan aspirasi melalui API dan masuk ke dalam basis data.

Lampiran 1 *Use case* deskripsi (lanjutan)

g Mengirim pesan status produksi

No	<i>Use case</i>	Deskripsi
7	Nama	Mengirim pesan status produksi
	Deskripsi	Pengguna dapat mengirimkan status produksi dari hasil panen rayanya kepada pemerintah. Hal ini bertujuan agar pemerintah dapat mengetahui status panen raya dari petani.
	Prekondisi	Pengguna berada di halaman utama dan memilih menu Status Produksi. Jika pengguna ingin mengirimkan status produksi panennya, maka pengguna memasukkan jenis komoditasnya, lokasi, dan jumlah produksinya.
	Proses	Pengguna mengirimkan status produksi panennya dengan mengisi <i>field</i> jenis komoditas, lokasi, dan jumlah produksinya,. Kemudian pengguna menekan tombol kirim.
	Kondisi Akhir	Aplikasi akan mengirimkan status produksi panen petani melalui API dan masuk ke dalam basis data.

h Melihat pesan opini masyarakat

No	<i>Use case</i>	Deskripsi
8	Nama	Melihat opini masyarakat
	Deskripsi	Pengguna melihat opini masyarakat melalui menu Timeline pada halaman utama. Pengguna dapat melihat informasi nama pengirimnya, tanggal, nama pasar, dan pesan opininya.
	Prekondisi	Pengguna berada di halaman utama dan memilih menu Timeline. Pengguna dapat melihat opini-opini yang masuk ke dalam basis data.
	Proses	Pengguna melihat opini masyarakat yang ada di menu Timeline. Pengguna dapat melihat opini yang ada dan dapat melihat opini dengan keterangan yang lebih jelas.
	Kondisi Akhir	Aplikasi akan menampilkan opini masyarakat dengan nama pengirim, tanggal, nama pasar, dan pesan opininya.

Lampiran 1 *Use case* deskripsi (lanjutan)

i Melihat pesan aspirasi petani

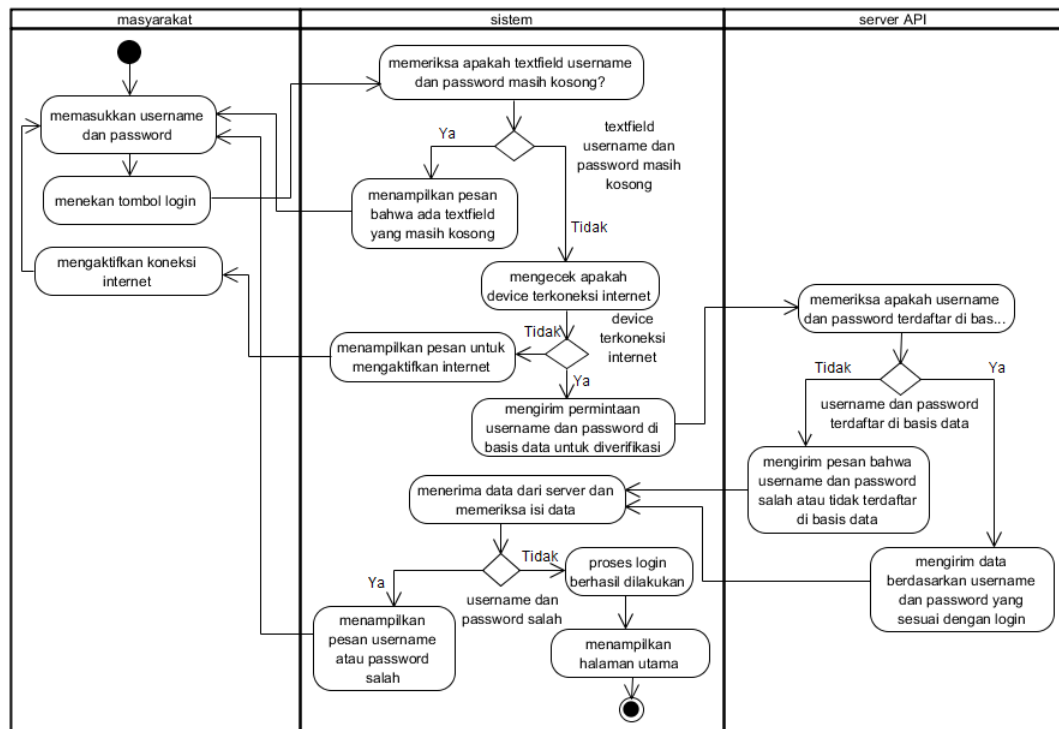
No	<i>Use case</i>	Deskripsi
9	Nama	Melihat aspirasi petani
	Deskripsi	Pengguna melihat aspirasi petani melalui menu <i>Timeline Aspirasi</i> pada halaman utama. Pengguna dapat melihat aspirasi-aspirasi yang dimasukkan oleh petani.
	Prekondisi	Pengguna berada di halaman utama dan memilih menu <i>Timeline Aspirasi</i> . Pengguna dapat melihat aspirasi yang masuk ke dalam basis data.
	Proses	Pengguna berada pada halaman utama dan memilih menu <i>Timeline Aspirasi</i> . Pengguna dapat melihat aspirasi petani dengan keterangan nama petani yang mengirimkan, subjek pesan, dan pesan aspirasinya.
	Kondisi Akhir	Aplikasi akan menampilkan aspirasi petani dengan keterangan nama petani, subjek pesan, dan pesan aspirasinya.

j Melihat status produksi panen

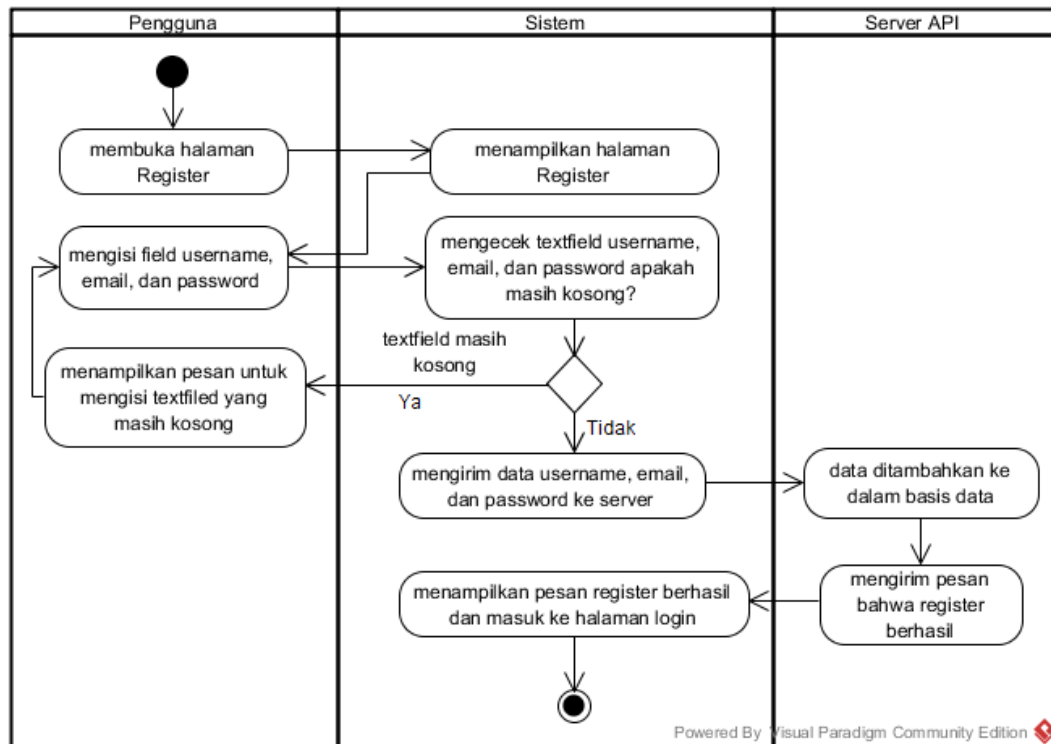
No	<i>Use case</i>	Deskripsi
10	Nama	Melihat status produksi
	Deskripsi	Pengguna melihat status produksi panen dari petani pada menu Notifikasi. Pengguna dapat melihat status panen suatu komoditas, lokasinya, dan jumlah produksinya.
	Prekondisi	Pengguna berada di halaman utama dan memilih menu Notifikasi. Pengguna dapat melihat status produksi petani yang masuk ke dalam basis data.
	Proses	Pengguna melihat informasi komoditas yang sedang panen, tanggal panen, jumlah komoditas yang panen beserta lokasinya.
	Kondisi Akhir	Aplikasi akan menampilkan status produksi panen dari petani dengan menampilkan nama petani, jenis komoditas, tanggal panen, lokasi panen, dan jumlah komoditasnya.

Lampiran 2 Activity diagram

a Login aplikasi Portal Harga

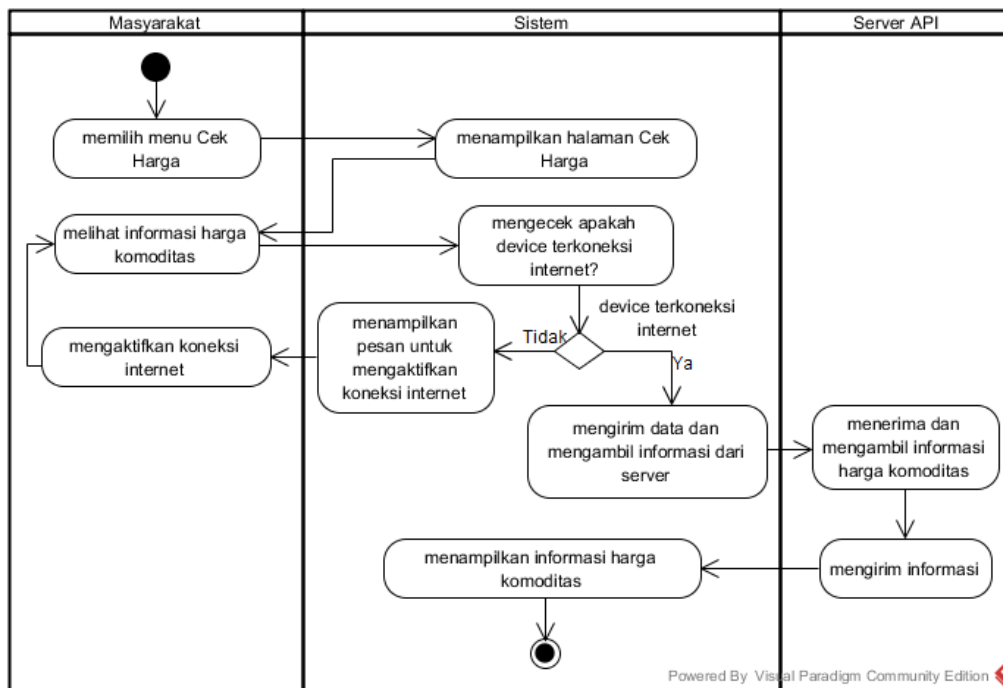


b Register

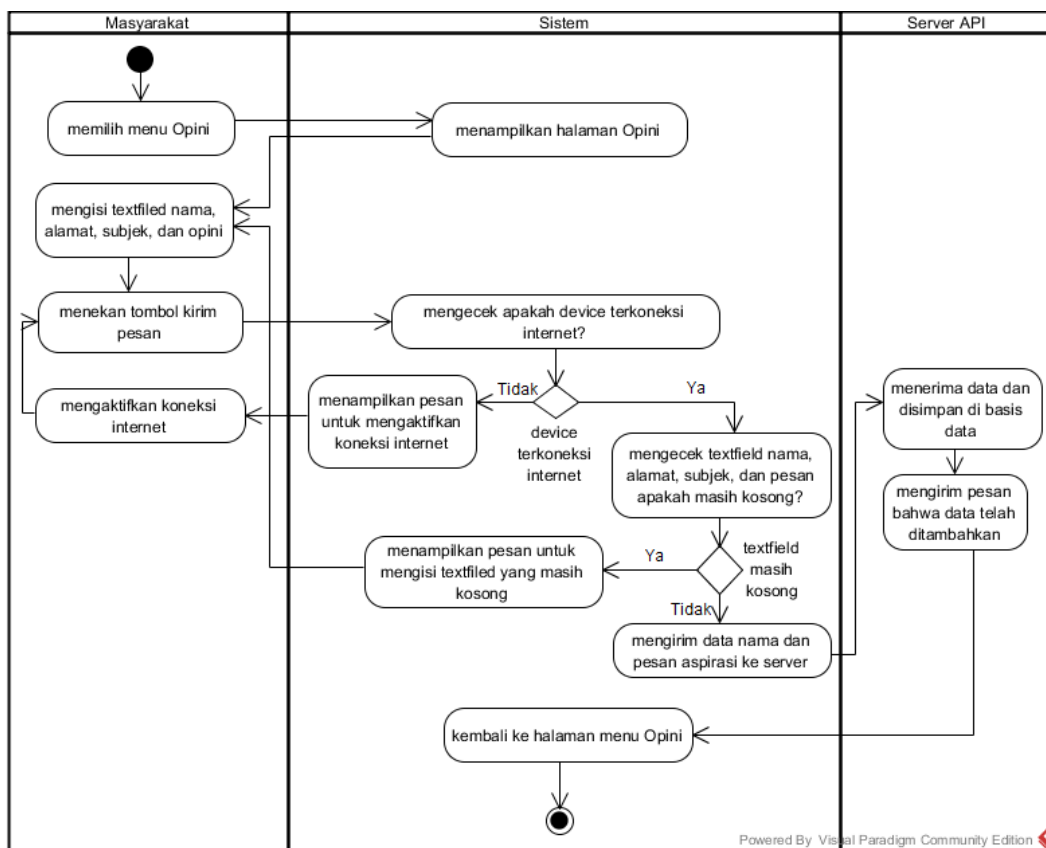


Lampiran 2 Activity diagram (lanjutan)

c Informasi harga aplikasi Portal Harga

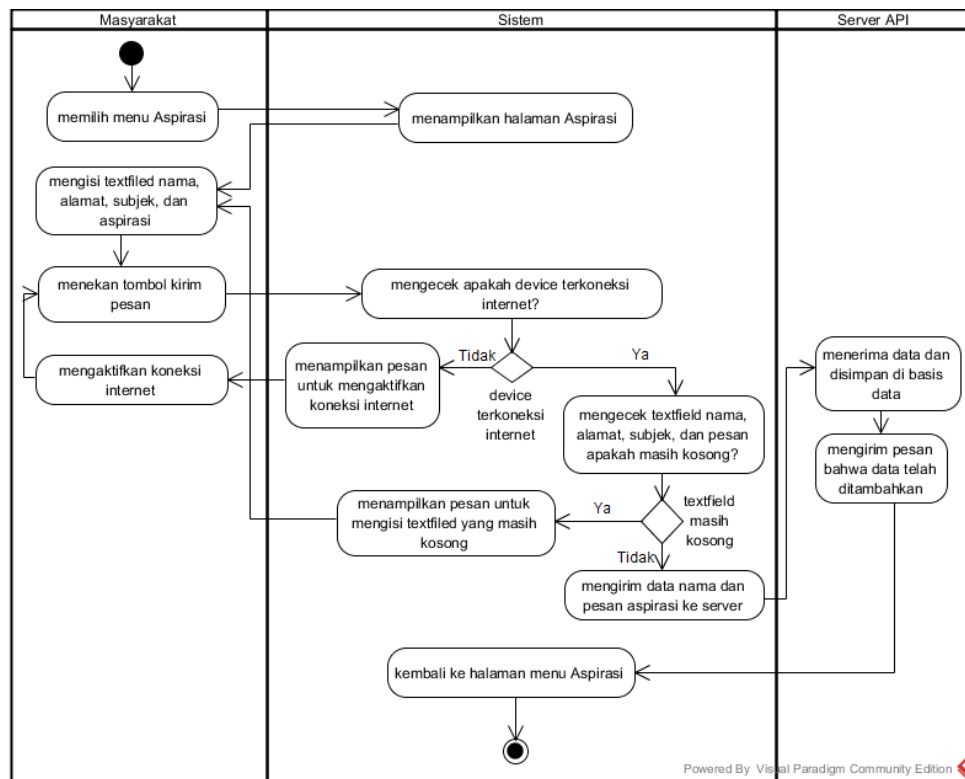


d Mengirim opini pada aplikasi Portal Harga

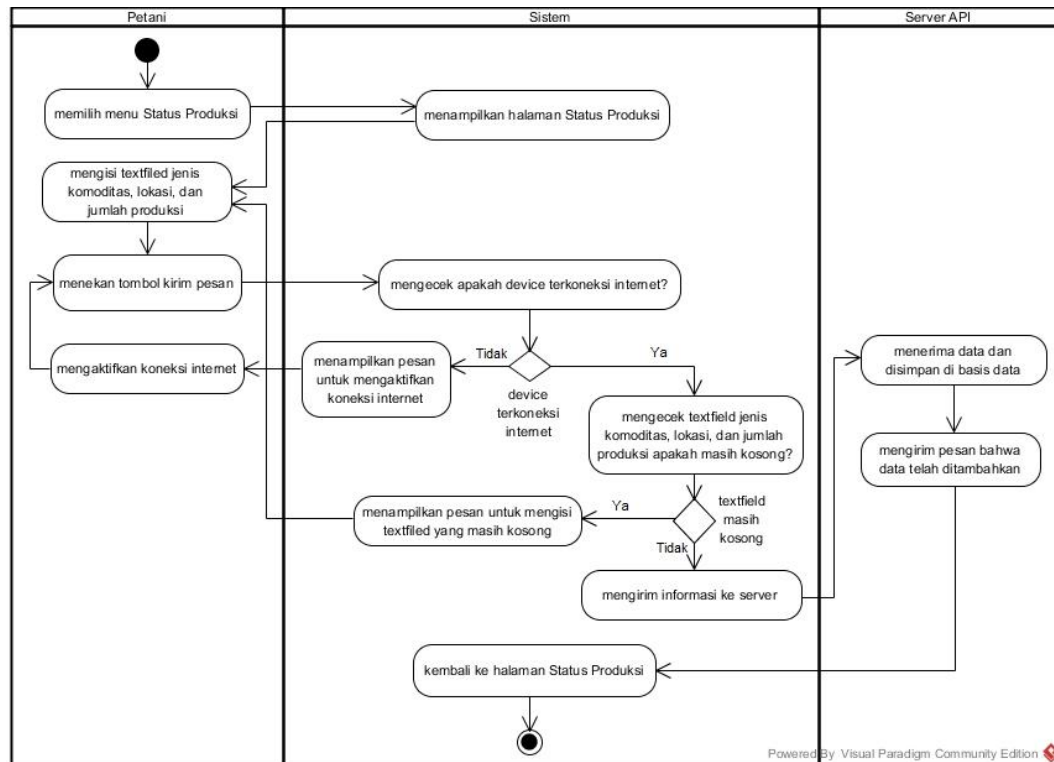


Lampiran 2 Activity diagram (lanjutan)

e Mengirim aspirasi petani pada aplikasi Portal Harga

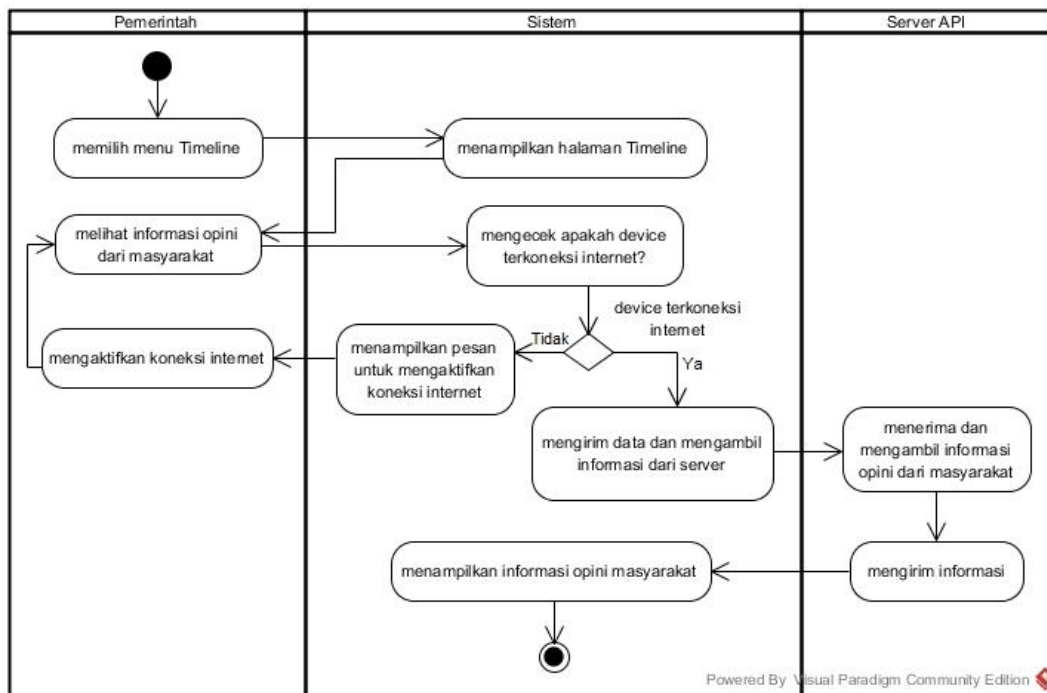


f Memberikan status produksi pada aplikasi Portal Harga

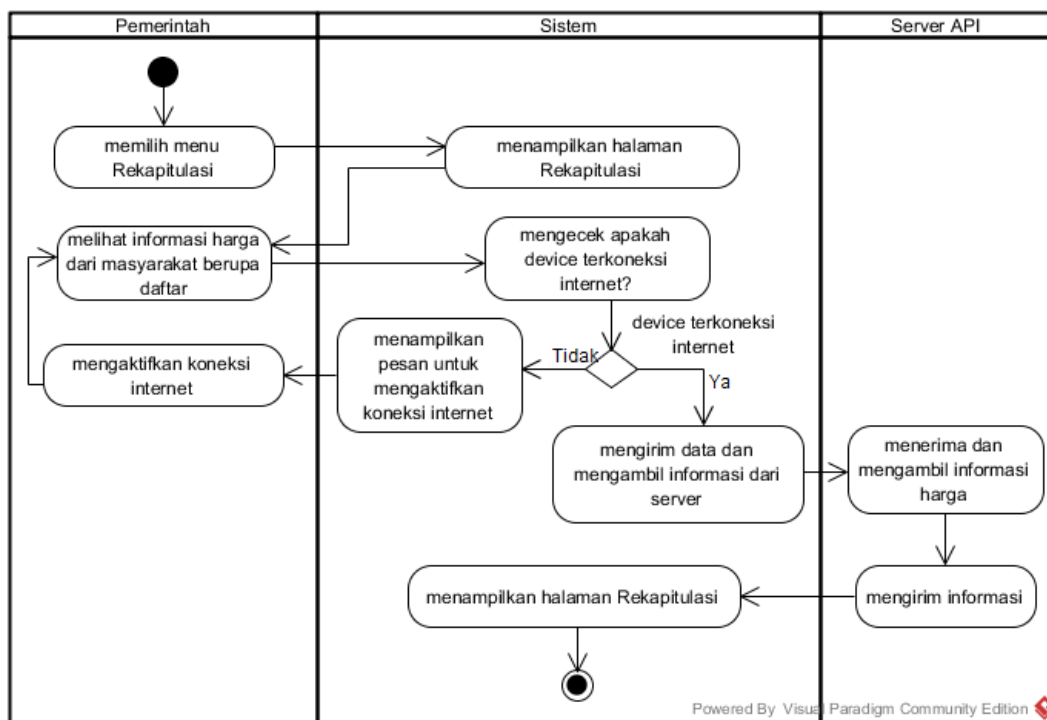


Lampiran 2 Activity diagram (lanjutan)

g Melihat opini dan aspirasi petani

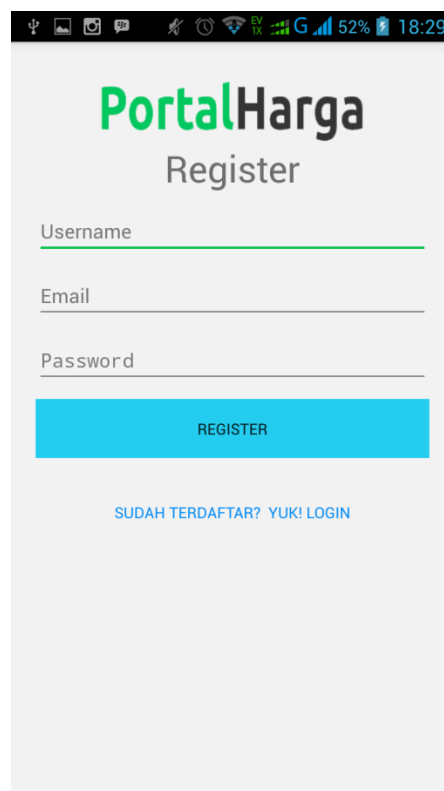
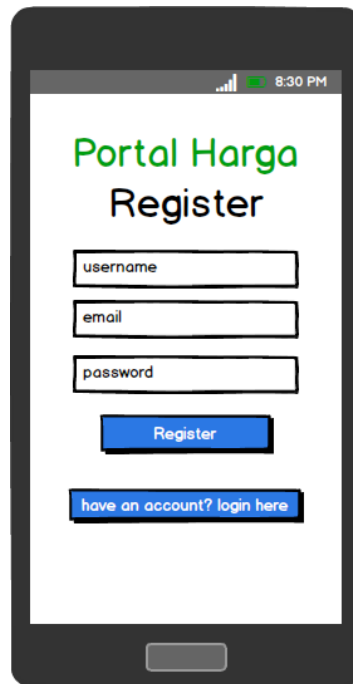


h Rekapitulasi informasi harga pada aplikasi Portal Harga

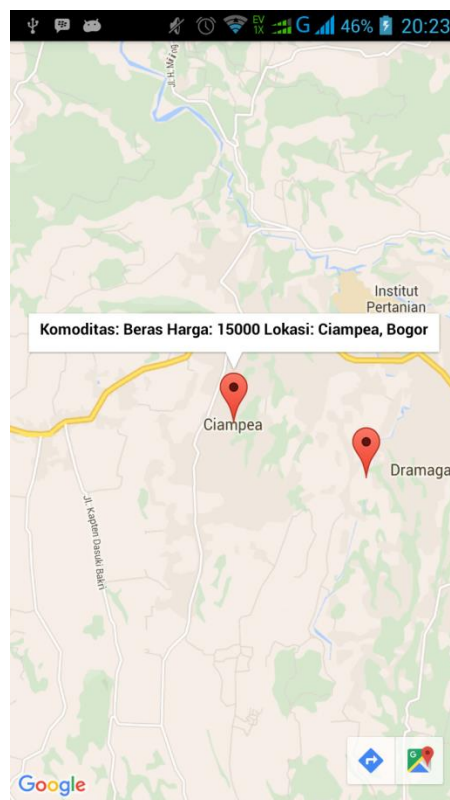


Lampiran 3 Rancangan antarmuka dan implementasi

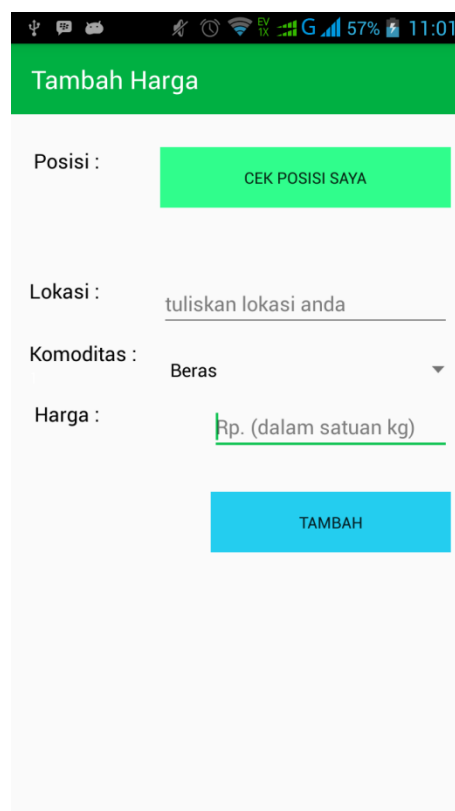
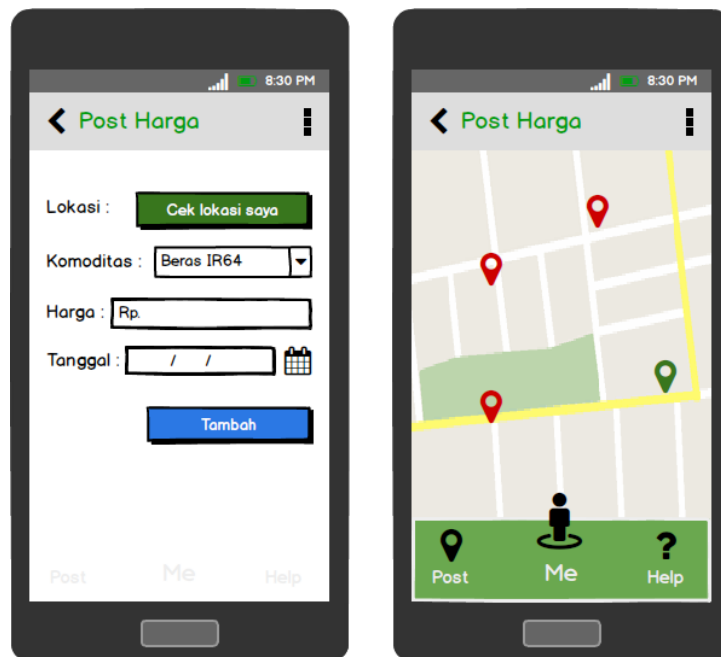
a Halaman *register*



Lampiran 3 Rancangan antarmuka dan implementasi (lanjutan)

b Halaman cek harga *maps*

Lampiran 3 Rancangan antarmuka dan implementasi (lanjutan)

c Halaman *post* harga

Lampiran 3 Rancangan antarmuka dan implementasi (lanjutan)

d Halaman opini operasi pasar

A wireframe illustration of a mobile application screen titled "Opini". The screen features a back arrow and the title "Opini" at the top. Below the title, there are three input fields: "Nama :", "Subjek :", and "Opini anda :". The "Opini anda :" field is a larger text area. At the bottom right, there is a blue button labeled "Kirim".

A screenshot of the implemented mobile application screen titled "Opini". The screen displays the following fields and elements:

- Nama :** A text input field containing the placeholder text "isi nama anda".
- Pasar :** A dropdown menu showing "Pasar Induk Kramat Jati".
- Subjek :** A text input field containing the placeholder text "isi subjek anda".
- Opini kamu :** A text input field containing the placeholder text "masukkan opini anda".
- KIRIM** A blue button at the bottom right.

Lampiran 3 Rancangan antarmuka dan implementasi (lanjutan)

e Halaman aspirasi petani

A wireframe illustration of a smartphone screen displaying the 'Aspirasi' (Aspirations) page. The status bar at the top shows signal strength, battery level, and the time 8:30 PM. The page has a header with a back arrow and the title 'Aspirasi'. Below the header, there are three input fields: 'Nama :', 'Subjek :', and 'Aspirasi anda :'. The 'Aspirasi anda :' field is a larger text area. At the bottom right, there is a blue button labeled 'Kirim'.

A screenshot of the 'Aspirasi' page on a smartphone, showing the implemented design. The status bar at the top displays various icons, signal strength, 52% battery, and the time 18:31. The page has a green header with the title 'Aspirasi'. Below the header, there are three input fields: 'Nama :', 'Subjek :', and 'Aspirasi kamu :'. The 'Aspirasi kamu :' field contains the placeholder text 'masukkan aspirasi anda'. At the bottom right, there is a blue button labeled 'KIRIM'.

Lampiran 3 Rancangan antarmuka dan implementasi (lanjutan)

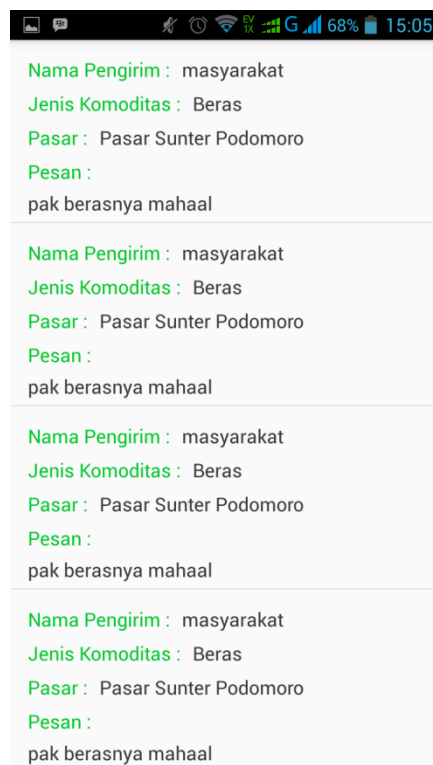
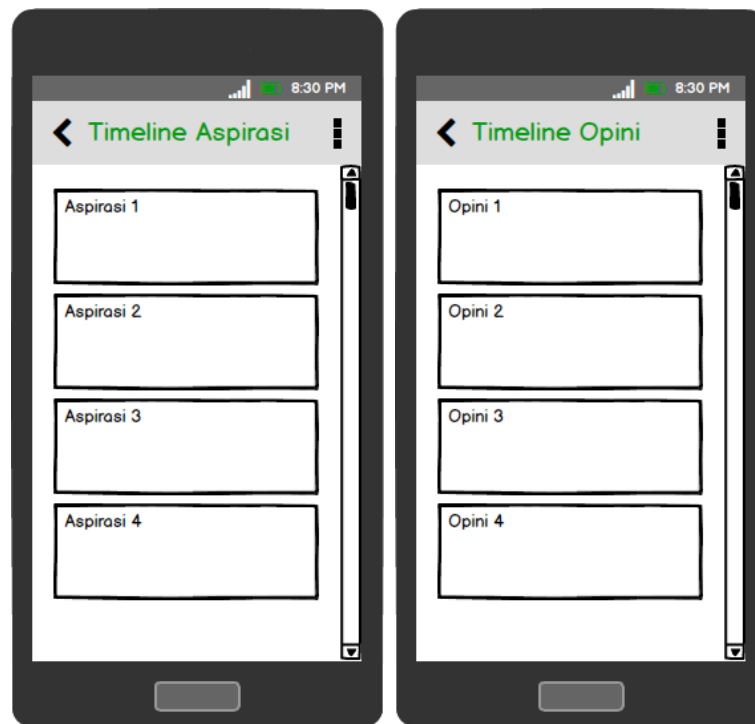
f Halaman status produksi

The image shows a mobile application interface for 'Status Produksi'. At the top, there is a back arrow and the title 'Status Produksi'. Below the title, there are several input fields: 'Nama' with a text box, 'Komoditas' with a dropdown menu showing 'Beras IR64', 'Lokasi' with a green button labeled 'Cek lokasi saya', 'Jumlah (kg)' with a text box, and 'Keterangan' with a text box. At the bottom, there is a blue button labeled 'Kirim'.

The image shows a mobile application interface for 'Status Produksi'. At the top, there is a green header with the title 'Status Produksi'. Below the header, there are several input fields: 'Komoditas' with a dropdown menu showing 'Bawang Merah', 'Lokasi' with a green button labeled 'CEK LOKASI SAYA', 'Jumlah produksi' with a text box containing 'dalam satuan kg', and 'Keterangan' with a text box containing 'keterangan status produksi anda'. At the bottom, there is a blue button labeled 'KIRIM'.

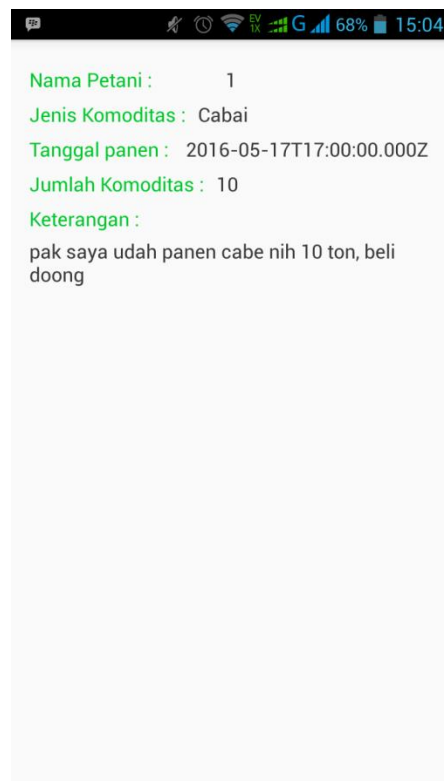
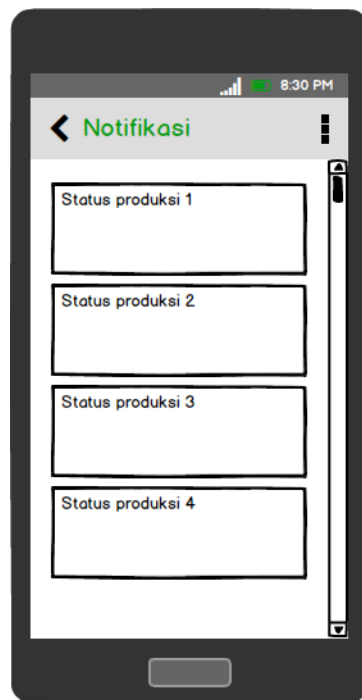
Lampiran 3 Rancangan antarmuka dan implementasi (lanjutan)

g Halaman *timeline*



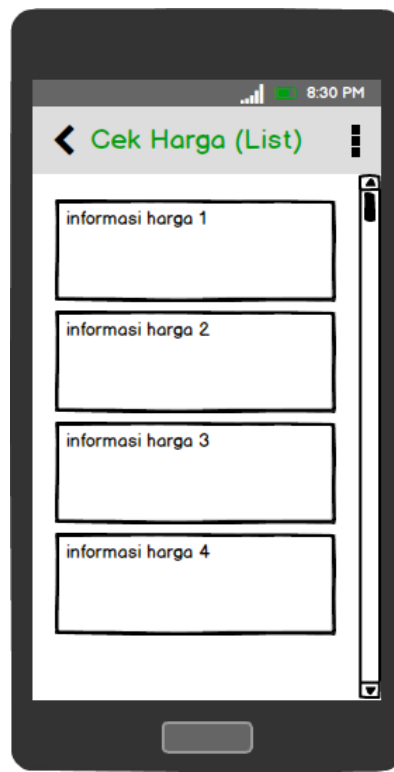
Lampiran 3 Rancangan antarmuka dan implementasi (lanjutan)

h Halaman notifikasi



Lampiran 3 Rancangan antarmuka dan implementasi (lanjutan)

i Halaman cek harga (*list*)



Nama Komoditas : Kedelai
Tanggal : 2016-06-04
Harga Komoditas : 12400
Lokasi : Lebak Bulus, Cilandak, South Jakarta City, Special Capital Region of Jakarta, Indonesia
Nama Komoditas : Kedelai
Tanggal : 2016-6-10
Harga Komoditas : 15000
Lokasi : Ciampea, Bogor, West Java, Indonesia
Nama Komoditas : Beras
Tanggal : didii
Harga Komoditas : 0
Lokasi : 30
Nama Komoditas : Beras
Tanggal : 2016-06-09
Harga Komoditas : 15000
Lokasi : Jalan Bungur Besar No. 169, Bungur, Senen, Bungur, Jakarta Pusat, Kota Jakarta Pusat, Daerah Khusus Ibukota Jakarta, Indonesia

Lampiran 4 Implementasi GET *parser*a Menu *timeline* aspirasi

```
106 String jsonStr = sh.makeServiceCall(url, ServiceHandler.GET);
107
108 Log.d("Response: ", "> " + jsonStr);
109 if (jsonStr != null) {
110     try {
111         JSONObject jsonObj = new JSONObject(jsonStr);
112         // Getting JSON Array node
113         aspirasi_data = jsonObj.getJSONArray(TAG_DATA);
114         // looping through All Contacts
115         for (int i = 0; i < aspirasi_data.length(); i++) {
116             JSONObject n = aspirasi_data.getJSONObject(i);
117             String id_aspirasi = n.getString(TAG_ID_ASPIRASI);
118             String nama = n.getString(TAG_NAMA);
119             String subject = n.getString(TAG_SUBJECT);
120             String time = n.getString(TAG_TIME);
121             String aspirasi_pesan = n.getString(TAG_ASPIRASI_PESAN);
122             // tmp hashmap for isi timeline
123             HashMap<String, String> data_aspirasi = new
                HashMap<String, String>();
124             // adding each child node to HashMap key => value
125             data_aspirasi.put(TAG_NAMA, nama);
126             data_aspirasi.put(TAG_SUBJECT, subject);
127             data_aspirasi.put(TAG_TIME, time);
128             data_aspirasi.put(TAG_ASPIRASI_PESAN, aspirasi_pesan);
129             // adding data timeline to timeline list
130             aspirasiList.add(data_aspirasi);
131         }
132     }
133 }
```

Lampiran 4 Implementasi GET *parser* (lanjutan)

b Menu cek harga (*list*)

```

109 String jsonStr = sh.makeServiceCall(url, ServiceHandler.GET);
111 Log.d("Response: ", "> " + jsonStr);
112 if (jsonStr != null) {
114     try {
115         JSONObject jsonObj = new JSONObject(jsonStr);
116         // Getting JSON Array node
118         data_rekap = jsonObj.getJSONArray(TAG_DATA);
119         // looping through All Contacts
121         for (int i = 0; i < data_rekap.length(); i++) {
122             JSONObject r = data_rekap.getJSONObject(i);
123             String id_komoditas = r.getString(TAG_ID_KOMODITAS);
125             String jenis = r.getString(TAG_JENIS);
126             String nama_komoditas = r.getString(TAG_NAMA_KOMODITAS);
127             String id_harga = r.getString(TAG_ID_HARGA);
128             String harga = r.getString(TAG_HARGA);
129             String tanggal = r.getString(TAG_TANGGAL);
130             String latitude = r.getString(TAG_LATITUDE);
131             String longitude = r.getString(TAG_LONGITUDE);
132             String lokasi = r.getString(TAG_LOKASI);
133             String id_user = r.getString(TAG_ID_USER);
134             // tmp hashmap for isi rekaplist
136             HashMap<String, String> data_rekap = new HashMap<String,
                String>();
137             // adding each child node to HashMap key => value
139             data_rekap.put(TAG_NAMA_KOMODITAS, nama_komoditas);
140             data_rekap.put(TAG_TANGGAL, tanggal);
141             data_rekap.put(TAG_HARGA, harga);
142             data_rekap.put(TAG_LOKASI, lokasi);
143             // adding data to rekap list
145             rekapList.add(data_rekap);
146         }

```

Lampiran 4 Implementasi GET *parser* (lanjutan)

c Menu notifikasi status produksi

```

113 String jsonStr = sh.makeServiceCall(url, ServiceHandler.GET);
114 Log.d("Response: ", "> " + jsonStr);
115 if (jsonStr != null) {
116     try {
117         JSONObject jsonObj = new JSONObject(jsonStr);
118         // Getting JSON Array node
119         notif_data = jsonObj.getJSONArray(TAG_DATA);
120         // looping through All Contacts
121         for (int i = 0; i < notif_data.length(); i++) {
122             JSONObject n = notif_data.getJSONObject(i);
123             String id_komoditas = n.getString(TAG_ID_KOMODITAS);
125             String jenis = n.getString(TAG_JENIS);
126             String id_infopetani = n.getString(TAG_ID_INFO_PETANI);
127             String id_user = n.getString(TAG_ID_USER);
128             String time = n.getString(TAG_TIME);
129             String longitude = n.getString(TAG_LONGITUDE);
130             String latitude = n.getString(TAG_LATITUDE);
131             String jumlah_komoditas = n.getString(TAG_JUMLAH_KOMODITAS);
132             String keterangan = n.getString(TAG_KETERANGAN);
133             String username = n.getString(TAG_USERNAME);
134             String password = n.getString(TAG_PASSWORD);
135             String nama = n.getString(TAG_NAMA);
136             String email = n.getString(TAG_EMAIL);
137             String hak_akses = n.getString(TAG_HAK_AKSES);
138             // tmp hashmap for isi timeline
139             HashMap<String, String> data_notif = new HashMap<String,
String>();
140             // adding each child node to HashMap key => value
141             data_notif.put(TAG_NAMA, nama);
142             data_notif.put(TAG_JENIS, jenis);
143             data_notif.put(TAG_TIME, time);
144             data_notif.put(TAG_JUMLAH_KOMODITAS, jumlah_komoditas);
145             data_notif.put(TAG_KETERANGAN, keterangan);
146             // adding data timeline to timeline list
147             notifList.add(data_notif);
148         }

```

Lampiran 5 Implementasi POST *parser*

a Menu *post* harga

```

163 private void kirimInformasi(){
164     final String id_komoditas = txtIdComm.getText().toString();
165     final String harga = txtHarga.getText().toString().trim();
166     if(harga.matches("")){
167         alert.showAlertDialog(MenuTambahHarga.this, "Tambah harga
            gagal", "Anda belum mengisi harga komoditas", false);
168     }
169 }
170
171 final String lokasi =
    txtLokasiPost.getText().toString().trim();
172 if(lokasi.matches("")){
173     alert.showAlertDialog(MenuTambahHarga.this, "Tambah harga
        gagal", "Anda belum mengisi lokasi anda", false);
174 }
175
176 StringRequest stringRequest = new
    StringRequest(Request.Method.POST, URL_TAMBAH_HARGA,
177 new Response.Listener<String>() {
178     @Override
179     public void onResponse(String response) {
180         Toast.makeText(MenuTambahHarga.this, response,
            Toast.LENGTH_LONG).show();
181     }
182 },
183 new Response.ErrorListener() {
184     @Override
185     public void onErrorResponse(VolleyError error) {
186         Toast.makeText(MenuTambahHarga.this,error.toString(),
            Toast.LENGTH_LONG).show();
187     }
188 }){
189     @Override
190     protected Map<String,String> getParams(){
191         Map<String,String> params = new HashMap<String, String>();
192         params.put(KEY_ID_KOMODITAS, id_komoditas);
193         params.put(KEY_HARGA, harga);
194         params.put(KEY_LOKASI, lokasi);
195         return params;
196     }
197 };

```

Lampiran 5 Implementasi POST *parser* (lanjutan)

b Menu opini

```

182 private void kirimOpini(){
183     final String id_user = txtIdUser.getText().toString();
184     final String id_pasar =
185         txtIdPasar.getText().toString().trim();
186     if(id_pasar.matches("")){
187         alert.showAlertDialog(MenuOpini.this, "Kirim opini gagal",
188             "Anda belum mengisi alamat pasar anda", false);
189     }
190     final String subjek =
191         txtSubjekOpini.getText().toString().trim();
192     if(subjek.matches("")){
193         alert.showAlertDialog(MenuOpini.this, "Kirim opini gagal",
194             "Anda belum mengisi subjek anda", false);
195     }
196     final String opini = txtOpini.getText().toString().trim();
197     if(opini.matches("")){
198         alert.showAlertDialog(MenuOpini.this, "Kirim opini gagal",
199             "Anda belum mengisi pesan opini anda", false);
200     }
201     StringRequest stringRequest = new
202         StringRequest(Request.Method.POST, URL_OPINI,
203             new Response.Listener<String>() {
204                 @Override
205                 public void onResponse(String response) {
206                     Toast.makeText(MenuOpini.this, response,
207                         Toast.LENGTH_LONG).show();
208                 }
209             },
210             new Response.ErrorListener() {
211                 @Override
212                 public void onErrorResponse(VolleyError error) {
213                     Toast.makeText(MenuOpini.this, error.toString(),
214                         Toast.LENGTH_LONG).show();
215                 }
216             }) {
217         @Override
218         protected Map<String,String> getParams(){
219             Map<String,String> params = new HashMap<String, String>();
220             params.put(KEY_ID_USER, id_user);
221             params.put(KEY_ID_PASAR, id_pasar);
222             params.put(KEY_SUBJEK, subjek);
223             params.put(KEY_PESAN, opini);
224             return params;
225         }
226     };

```

Lampiran 5 Implementasi POST *parser* (lanjutan)

c Menu status produksi

```

162 private void kirimStatus(){
163     final String id_komoditas =
        txtIdComm.getText().toString().trim();
164
165     final String jumlah_komoditas =
        txtProduksi.getText().toString().trim();
166     if(jumlah_komoditas.matches("")){
167         alert.showAlertDialog(MenuStatusProduksi.this, "Kirim
            Status Produksi gagal", "Anda belum mengisi jumlah produksi
            anda", false);
168     }
169
170     final String keterangan =
        txtKeterangan.getText().toString().trim();
171     if(keterangan.matches("")){
172         alert.showAlertDialog(MenuStatusProduksi.this, "Kirim
            Status Produksi gagal", "Anda belum mengisi keterangan
            anda", false);
173     }
174
175     StringRequest stringRequest = new
        StringRequest(Request.Method.POST, URL_STATUS,
176         new Response.Listener<String>() {
177             @Override
178             public void onResponse(String response) {
179                 Toast.makeText(MenuStatusProduksi.this, response,
                    Toast.LENGTH_LONG).show();
180             }
181         },
182         new Response.ErrorListener() {
183             @Override
184             public void onErrorResponse(VolleyError error) {
185                 Toast.makeText(MenuStatusProduksi.this,error.toString(),
                    Toast.LENGTH_LONG).show();
186             }
187         }) {
188         @Override
189         protected Map<String,String> getParams(){
190             Map<String,String> params = new HashMap<String, String>();
191             params.put(KEY_ID_KOMODITAS,id_komoditas);
192             params.put(KEY_JUMLAH_KOMODITAS, jumlah_komoditas);
193             params.put(KEY_KETERANGAN, keterangan);
194             return params;
195         }
196     };

```

Lampiran 6 Proses pengujian aplikasi Portal Harga

Nama bagian	Skenario	Hasil yang diharapkan	Status pengujian
Masyarakat			
Login	Terdapat <i>field</i> yang belum terisi	Menampilkan pesan <i>field</i> harus diisi dengan lengkap	Berhasil
	Mengisi <i>field username</i> dan <i>password</i> dengan benar	Masuk ke halaman utama	Berhasil
Proses menambahkan informasi harga	Terdapat <i>field</i> yang belum terisi	Data tidak tersimpan, kembali ke tampilan <i>field</i> yang kosong dan menampilkan pesan eror	Berhasil
Proses melihat informasi harga	Mengisi <i>field</i> secara lengkap	Data berhasil tersimpan melalui API	Berhasil
	Memilih menu Cek Harga yang berisi halaman Google Maps	Google Maps berhasil terbuka dan menunjukkan informasi lokasi harga	Berhasil
Proses memasukkan opini <i>request</i> pasar	Terdapat <i>field</i> yang belum terisi	Data tidak tersimpan, kembali ke tampilan <i>field</i> yang kosong dan menampilkan pesan eror	Berhasil
	Mengisi <i>field</i> yang tersedia dengan lengkap	Data berhasil tersimpan melalui API	Berhasil
Petani			
Login	Terdapat <i>field</i> yang belum terisi	Menampilkan pesan <i>field</i> harus diisi dengan lengkap	Berhasil
	Mengisi <i>field username</i> dan <i>password</i> dengan benar	Masuk ke halaman utama	Berhasil
Proses memasukkan aspirasi petani	Terdapat <i>field</i> yang belum terisi	Data tidak tersimpan, kembali ke tampilan <i>field</i> yang kosong dan menampilkan pesan eror	Berhasil
	Mengisi <i>field</i> yang tersedia dengan lengkap	Data berhasil tersimpan melalui API	Berhasil

Lampiran 6 Proses pengujian aplikasi Portal Harga (lanjutan)

Nama bagian	Skenario	Hasil yang diharapkan	Status pengujian
Proses memasukkan status produksi panen	Terdapat <i>field</i> yang belum terisi	Data tidak tersimpan, kembali ke tampilan <i>field</i> yang kosong dan menampilkan pesan eror	Berhasil
	Mengisi <i>field</i> yang tersedia dengan lengkap	Data berhasil tersimpan melalui API	Berhasil
Pemerintah			
Login	Terdapat <i>field</i> yang belum terisi	Menampilkan pesan <i>field</i> harus diisi dengan lengkap	Berhasil
	Mengisi <i>field username</i> dan <i>password</i> dengan benar	Masuk ke halaman utama	Berhasil
Proses melihat informasi harga melalui Google Maps	Memilih menu Cek Harga yang berisi halaman Google Maps	Google Maps berhasil terbuka dan menunjukkan informasi lokasi harga	Berhasil
Proses melihat informasi harga melalui list	Memilih menu Cek Harga yang berisi halaman list informasi harga	Menampilkan list berisi informasi harga	Berhasil
Proses melihat opini <i>request</i> pasar	Memilih menu <i>timeline</i> opini	Menampilkan list berisi opini-opini <i>request</i> pasar dari masyarakat	Berhasil
Proses melihat aspirasi petani	Memilih menu <i>timeline</i> aspirasi	Menampilkan list berisi aspirasi-aspirasi dari petani	Berhasil
Proses melihat status produksi panen raya	Memilih menu notifikasi	Menampilkan list berisi status produksi panen dari petani	Berhasil

RIWAYAT HIDUP

Penulis dilahirkan di Jakarta pada tanggal 12 Desember 1994 yang merupakan anak kedua dari enam bersaudara dengan Ayah bernama Drs. Dibyو Hutomo dan (Almh) Ibu bernama Dra. Dewi Handayani. Pada tahun 2009, penulis menempuh pendidikan menengah atas di SMA Negeri 11 Tangerang dan masuk program IPA. Pada tahun 2012, penulis lulus dari SMA Negeri 11 Tangerang dan diterima di Departemen Ilmu Komputer, FMIPA, Institut Pertanian Bogor melalui jalur SNMPTN Undangan di tahun yang sama.

Pada masa perkuliahan, penulis aktif menjadi asisten praktikum mata kuliah Algoritme dan Pemrograman, Metode Kuantitatif, dan Penerapan Komputer. Pada tahun 2014, penulis aktif berorganisasi di Himpunan Mahasiswa Ilmu Komputer (Himalkom) Divisi HRD sebagai staf. Pada pertengahan tahun 2015, penulis melaksanakan kegiatan praktik kerja lapangan (PKL) di Balai Besar Industri Agro (BBIA) Bogor.