

**PENGEMBANGAN MODUL PELAPORAN HARGA KOMODITAS  
PERTANIAN PADA APLIKASI PORTAL HARGA UNTUK  
MASYARAKAT MENGGUNAKAN REST API**

**KADEK GRIHADEVI KARDINI**



**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 Modul Pelaporan Harga Komoditas Pertanian pada Aplikasi Portal Harga untuk Masyarakat Menggunakan REST APIs 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

*Kadek Grihadevi Kardini*  
NIM G64120023

## ABSTRAK

KADEK GRIHADEVI KARDINI. Pengembangan Modul Pelaporan Harga Komoditas Pertanian pada Aplikasi Portal Harga untuk Masyarakat Menggunakan REST APIs. Dibimbing oleh DEAN APRIANA RAMADHAN.

Harga komoditas pertanian di beberapa pasar di Indonesia sering mengalami perubahan dari waktu ke waktu. Kelangkaan komoditas pertanian sering terjadi akibat kesenjangan antara penawaran dan permintaan di pasar. Keterlambatan penyampaian informasi hasil panen, khususnya saat panen raya dari petani ke pemerintah merupakan salah satu faktor pemicu kenaikan harga dan kelangkaan komoditas pertanian. Kehadiran tengkulak juga menyebabkan ketidakstabilan harga di pasar dan membawa kerugian bagi petani dan masyarakat. Hasil panen petani dibayar dengan harga sangat murah sedangkan harga di pasar meningkat sehingga masyarakat mendapatkan harga yang mahal. Penelitian ini berupa pengembangan aplikasi web tentang harga komoditas pertanian yang ada di pasar, yaitu beras, daging merah, cabai, kedelai, dan bawang merah menggunakan *REpresentational State Transfer* (REST) *Application Programming Interface* (API) dengan metode *Agile Extreme Programming*. Penelitian ini berhasil mengembangkan aplikasi web berbasis REST APIs yang dapat mempermudah masyarakat dalam mengetahui harga komoditas, menambahkan informasi harga komoditas, dan meminta pemerintah melaksanakan operasi pasar untuk mencapai kestabilan harga komoditas pertanian.

Kata kunci: *agile XP*, harga, kesenjangan, komoditas pertanian, REST APIs

## ABSTRACT

KADEK GRIHADEVI KARDINI. Development of Agricultural Commodities Price Reporting Module in Portal Harga Application for Public Using REST APIs. Supervised by DEAN APRIANA RAMADHAN.

Agricultural commodities price in some markets in Indonesia often changes over time. Scarcity of agricultural commodities often caused by the gap between supply and demand in the market. Delays on delivery of yields information, especially during harvest from farmers to government is one of the factors that triggering price increases and scarcity of agricultural commodities. The presence of middleman also cause price volatility in the market and bring suffered losses to farmers and public. The yields were paid in a very low price while price in the market get increased so that public get an expensive price. This research is the development of web-based application on the price of agricultural commodities on the market, such as rice, meat, chili, soybean, and onion using *Representational State Transfer* (REST) *Application Programming Interface* (API) with *Agile Extreme Programming* methods. This research succesfully developed web application using REST APIs that facilitate the public in knowing the price of commodities, adding commodity price information, and ask the government to carry out market operations to achieve stability in agricultural commodity prices.

Keywords: *agile XP*, agricultural commodities, gap, price, REST APIs

**PENGEMBANGAN MODUL PELAPORAN HARGA KOMODITAS  
PERTANIAN PADA APLIKASI PORTAL HARGA UNTUK  
MASYARAKAT MENGGUNAKAN REST API**

**KADEK GRIHADEVI KARDINI**

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**



Judul Skripsi: Pengembangan Modul Pelaporan Harga Komoditas Pertanian pada  
Aplikasi Portal Harga untuk Masyarakat Menggunakan REST APIs  
Nama : Kadek Grihadevi Kardini  
NIM : G64120023

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 Tuhan Yang Maha Esa atas segala karunia-Nya sehingga skripsi ini berhasil diselesaikan. Tema yang dipilih dalam penelitian yang dilaksanakan sejak bulan Desember 2015 ini ialah pelaporan harga, dengan judul Pengembangan Modul Pelaporan Harga Komoditas Pertanian pada Aplikasi Portal Harga untuk Masyarakat Menggunakan REST APIs. Dalam penyusunan skripsi ini banyak pihak yang telah membantu dan mendukung penulis dalam hal apapun. Oleh karena itu, penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

- 1 Dosen pembimbing skripsi, Bapak Dean Apriana Ramadhan, SKomp, Mkom yang selalu memberikan dukungan dan arahan selama penelitian dan penulisan skripsi ini.
- 2 Dosen penguji skripsi, Bapak Firman Ardiansyah, Skom, MSi dan Ibu Dr Yani Nurhadryani, SSi, MT yang telah banyak memberikan saran.
- 3 Keluarga penulis, Ibu Ni Wayan Sri Astuti, Bapak I Ketut Pegeg, Kakak Putu Deva Kari Kardika, dan Adik Nyoman Diva Wedantara atas doa, dukungan, serta semangat positif yang terus diberikan untuk penulis.
- 4 Teman-teman satu bimbingan, Reza Fahlevi, Dian Setiana, Dieva Setyatama, Indriyani, Moh Reza, dan Shellafuri atas dukungan serta bantuan selama penelitian berlangsung.
- 5 Sahabat-sahabat penulis, Anom Sukmayana, Juananda Dwi, Hilmi Salim, Muhammad Fariz, dan Alanikika Pratyaksa atas bantuan dan semangat bagi penulis.
- 6 Teman-teman seperjuangan Ilmu Komputer angkatan 49 atas kebersamaan, bantuan, serta semangat bagi penulis.

Semoga karya ilmiah ini bermanfaat dan memberikan kontribusi nyata terhadap perkembangan ilmu pengetahuan.

Bogor, Agustus 2016

*Kadek Grihadevi Kardini*



## 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	2
TINJAUAN PUSTAKA	3
Rest APIs	3
METODE	3
Data Penelitian	3
Arsitektur Penelitian	4
Tahapan Penelitian	4
Perangkat Penelitian	5
HASIL DAN PEMBAHASAN	6
Iterasi Pertama	6
Iterasi Kedua	12
SIMPULAN DAN SARAN	16
Simpulan	16
Saran	16
DAFTAR PUSTAKA	16
LAMPIRAN	18
RIWAYAT HIDUP	26

## DAFTAR TABEL

1	<i>User story</i> Portal Harga	6
2	<i>Fully developed use case description</i> fungsi <i>post</i> harga	7
3	Pengujian fungsi REST iterasi pertama	12
4	Pengujian fungsi REST iterasi kedua	16

## DAFTAR GAMBAR

1	Tahapan pengembangan model <i>Extreme Programming</i> (Pressman 2010)	4
2	Arsitektur penelitian aplikasi Portal Harga	6
3	<i>Use case diagram</i> Portal Harga dari sisi pengguna masyarakat	7
4	<i>Swimlane diagram</i> fungsi <i>post</i> harga	8
5	<i>Mockup</i> halaman <i>post</i> harga	9
6	Basis data aplikasi Portal Harga	9
7	Kode program untuk menghubungkan REST dengan basis data	10
8	Kode program REST API untuk fungsi <i>post</i> harga	11
9	Kode program REST API untuk fungsi cek harga	11
10	Data teks JSON hasil dari fungsi cek harga	12
11	<i>Swimlane diagram</i> fungsi aspirasi	13
12	Tampilan halaman <i>post</i> harga pada web	14
13	Kode program penambahan token OAuth	14
14	Kode program REST API untuk fungsi aspirasi	15
15	Kode <i>script</i> implementasi fungsi <i>post</i> harga	15

## DAFTAR LAMPIRAN

1	<i>Fully developed use case description</i> fungsi melihat harga komoditas (cek harga)	18
2	<i>Fully developed use case description</i> fungsi melakukan <i>request</i> operasi pasar (aspirasi)	19
3	<i>Swimlane diagram</i> fungsi cek harga	20
4	<i>Mockup</i> fungsi cek harga	20
5	<i>Mockup</i> fungsi aspirasi	21
6	Tampilan halaman beranda	21
7	Tampilan halaman cek harga	22
8	Tampilan halaman aspirasi	23
9	Kode html halaman <i>post</i> harga	24

# PENDAHULUAN

## Latar Belakang

Indonesia sebagai negara agraris memiliki sektor pertanian yang menjadi penggerak utama dalam kehidupan masyarakat. Undang-Undang Republik Indonesia Nomor 19 Tahun 2013 tentang Perlindungan dan Pemberdayaan Petani menyatakan bahwa para petani diberikan jaminan perlindungan guna membantu petani dalam menghadapi permasalahan kesulitan memperoleh sarana dan prasarana produksi, kepastian usaha, risiko harga, kegagalan panen, praktek ekonomi biaya tinggi, dan perubahan iklim. Namun pada prakteknya komoditas pertanian yang dihasilkan oleh petani khususnya saat panen raya sering kali membawa kerugian besar terhadap petani itu sendiri akibat keterlambatan penyampaian informasi dari petani ke pemerintah. Akibatnya pemerintah terlambat dalam membeli hasil panen, sehingga didahului oleh pedagang pengumpul atau tengkulak yang membeli hasil panen dengan harga seminimal mungkin. Hal ini juga menyebabkan kerugian pada masyarakat akibat kenaikan harga yang disebabkan oleh kesenjangan antara penawaran dan permintaan di pasar. Kondisi inilah yang menjadi kesalahpahaman antara petani dan pemerintah, serta pemerintah dan masyarakat.

Seiring perkembangan teknologi dan informasi yang kian pesat, masyarakat dituntut untuk melakukan perubahan pada pola komunikasi dan bisnis yang dilakukan. Hal ini terlihat dari banyaknya jumlah pengguna telepon pintar (*smartphone*) yang dapat mengakses beragam informasi secara cepat dan *real time* melalui internet. Berdasarkan penelitian APJII (2014) pengguna internet di Indonesia sebesar 38.8% berusia 26 sampai 35 tahun dan didominasi oleh masyarakat yang berjenis kelamin perempuan. Pada usia ini, masyarakat khususnya perempuan cenderung lebih banyak terjun langsung ke pasar dan memperhatikan harga komoditas di pasar. Sebuah aplikasi informasi harga pertanian berbasis *mobile* dapat menjadi salah satu solusi dalam penyampaian informasi dari dan untuk petani, pemerintah, dan masyarakat.

Salah satu aplikasi mengenai informasi harga komoditas pertanian yang telah dibangun sebelumnya yaitu aplikasi Pantau Harga pada tahun 2015 oleh Coding Stelsel. Pada aplikasi ini terdapat fitur cek harga, *post* harga, dan jual komoditas. Target pengguna dari aplikasi ini yaitu masyarakat, KUD, dan petani. Ada pula penelitian yang membangun aplikasi informasi harga produk pertanian berbasis SMS menggunakan metode SDLC yang diperuntukkan bagi petani dan pelaku jual beli hasil pertanian (Astika *et al* 2009). Pada penelitian ini ditambahkan fitur yang belum terdapat pada aplikasi Pantau Harga yaitu fitur aspirasi atau *request* operasi pasar kepada pemerintah yang bertujuan untuk menanggulangi kelangkaan komoditas di pasar. Adapun metode yang digunakan pada penelitian ini ialah metode *Agile Extreme Programming* untuk mempermudah dalam mengatasi perubahan-perubahan kebutuhan secara cepat. REST APIs digunakan karena lebih sederhana, tidak memerlukan struktur data khusus, serta kuat dan *scalable* karena hanya mentransfer data. Aliran informasi yang cepat dan tanggapan yang tepat diharapkan mampu mengatasi kesenjangan pada penawaran dan permintaan terhadap komoditas pertanian, juga mencapai kestabilan harga komoditas pertanian.

### **Perumusan Masalah**

Rumusan permasalahan pada penelitian ini yaitu:

- 1 Bagaimana mengembangkan aplikasi web pelaporan harga komoditas pertanian berbasis REST APIs?
- 2 Bagaimana mengembangkan aplikasi pelaporan harga komoditas pertanian yang mampu memudahkan masyarakat dalam memilih komoditas dengan harga yang sesuai kebutuhan masyarakat?
- 3 Bagaimana melibatkan masyarakat dalam pemantauan harga di pasar sehingga informasi tentang harga dapat sampai kepada *stakeholder* terkait secara cepat?

### **Tujuan Penelitian**

Tujuan dari penelitian ini adalah:

- 1 Mengembangkan aplikasi web pelaporan harga komoditas pertanian berbasis REST APIs untuk masyarakat.
- 2 Memudahkan masyarakat memilih komoditas pertanian dengan harga yang sesuai kebutuhan.
- 3 Melibatkan masyarakat dalam memantau harga komoditas pertanian di pasar dengan memasukkan data harga komoditas pertanian dan meminta pelaksanaan operasi pasar kepada pemerintah.

### **Manfaat Penelitian**

Manfaat yang diperoleh dari penelitian ini adalah:

- 1 Masyarakat memperoleh informasi secara cepat terkait harga komoditas pertanian di pasar sehingga bisa memilih komoditas dengan harga yang sesuai kebutuhan.
- 2 Masyarakat dapat berperan aktif dalam melakukan pemantauan harga di pasar.

### **Ruang Lingkup Penelitian**

Lingkup dari penelitian ini, yaitu:

- 1 Penelitian ini dibatasi pada komoditas pertanian yang ada di pasar yaitu beras, daging sapi, cabai (cabai merah besar, cabai merah keriting, cabai rawit merah, dan cabai rawit hijau), kedelai, dan bawang merah.
- 2 Penelitian dilakukan dalam lingkup pasar induk di wilayah Jabodetabek (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).
- 3 Penelitian ini berfokus pada upaya pengumpulan data dengan mengasumsikan data yang dimasukkan oleh pengguna merupakan data yang valid.

## TINJAUAN PUSTAKA

### Rest APIs

*REpresentational State Transfer* (REST) merupakan suatu jenis *web service* yang menerapkan konsep perpindahan *state*. *State* dapat digambarkan seperti ketika *browser* meminta suatu halaman *web*, *server* mengirimkan *state* halaman *web* yang sekarang ke *browser* (Fauziah 2013). Untuk melakukan suatu aktivitas, REST bekerja dengan bernavigasi melalui *link-link* HTTP, seolah terjadi perpindahan antara *state* satu dengan yang lain. *Service* yang digunakan mengambil *method* milik HTTP yaitu GET, POST, PUT, atau DELETE.

Model arsitektur REST didasari oleh empat prinsip utama teknologi, yaitu: (1) *resource identification through URI*, RESTful *web service* mencari sekumpulan sumber daya yang mengidentifikasi interaksi antar klien; (2) *uniform interface*, sumber daya yang dimanipulasi *Create*, *Read*, *Update*, dan *Delete* (CRUD) menggunakan operasi GET, POST, PUT, dan DELETE; (3) *self-descriptive message*, sumber daya tidak terikat sehingga dapat mengakses berbagai format konten, seperti HTML, XML, *plain text*, PDF, JPEG, meta data, dan lainnya; serta (4) *stateful interactions through hyperlinks*, interaksi dengan sumber daya bersifat *stateless* (2008).

Keunggulan dari penggunaan arsitektur REST yaitu kesederhanaan konsep, ketahanan terhadap perubahan, dan keseragaman. Arsitektur REST menjadi lebih sederhana karena menggunakan format standar (HTTP, HTML, XML, URI, dan lainnya). Namun jika diperlukan proses pengambilan data, konten berupa teks dari hasil eksekusi *web service* dapat diolah dalam format teks seperti XML atau HTML menggunakan utilitas komunikasi data melalui koneksi *socket* protokol HTTP (Istiyanto 2012).

API merupakan *software interface* yang terdiri dari kumpulan instruksi yang disimpan dalam bentuk library dan menjelaskan bagaimana suatu software dapat berinteraksi dengan software lainnya. Secara struktural, API merupakan spesifikasi dari suatu data structure, objects, functions, beserta parameter-parameter yang diperlukan untuk mengakses resource dari aplikasi tersebut. Seluruh spesifikasi membentuk suatu interface yang dimiliki oleh aplikasi untuk berkomunikasi dengan aplikasi lain. API juga dapat digunakan dalam berbagai bahasa pemrograman maupun menggunakan URL yang telah disediakan suatu *website*.

## METODE

### Data Penelitian

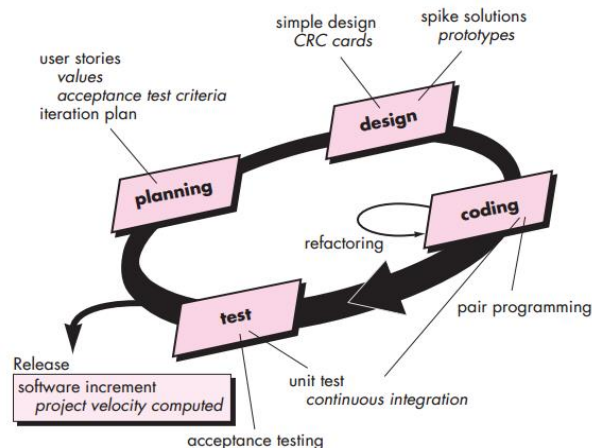
Data yang digunakan pada penelitian ini adalah data yang bersumber dari pengguna aplikasi yang memasukkan langsung informasi mengenai jenis komoditas, harga, dan lokasi penjualan, serta data *request* operasi pasar yang dikirim masyarakat ke pemerintah.

## Perancangan Arsitektur Penelitian

Arsitektur yang digunakan pada penelitian ini yaitu MySQL sebagai basis data. Bahasa pemrograman Node.js digunakan untuk membangun REST API. Bahasa pemrograman PHP dengan *framework* Code Igniter (CI) 3.03 digunakan untuk membuat tampilan pada halaman web. *Google Maps* API digunakan untuk menandai lokasi dengan mengambil *longitude* dan *latitude*. Keseluruhan sistem dibangun berbasis web yang ditujukan kepada masyarakat.

## Tahapan Penelitian

Adapun metode yang digunakan pada penelitian ini yaitu *Extreme Programming* (XP) yang merupakan bagian dari *Agile Software Engineering*. XP dipilih untuk menangani perubahan-perubahan yang biasanya terjadi pada saat pengembangan sudah hampir berakhir (Widodo 2008). Tahapan-tahapan yang dilakukan pada penelitian ini yaitu perencanaan, desain, pengkodean, dan pengujian. Gambar 1 menunjukkan tahapan pengembangan sistem menggunakan XP menurut Pressman (2010).



Gambar 1 Tahapan pengembangan model *Extreme Programming* (Pressman 2010)

## Perencanaan (XP Planning)

Tahap perencanaan dimulai dengan pengumpulan kebutuhan-kebutuhan pengguna atau *user requirement*. Setelah semua kebutuhan terkumpul, kebutuhan-kebutuhan ini kemudian dianalisis untuk mengetahui seluk beluk aplikasi yang akan dibangun, salah satunya dengan membuat *use case diagram*, *user story*, dan *fully developed use case description*. *Use case diagram* dibuat untuk menggambarkan fitur-fitur utama sistem dan interaksi antara aktor dengan sistem yang dibangun. Pembuatan *user story* bertujuan untuk mempermudah *developer* dalam memahami keinginan dan kebutuhan pengguna, sehingga akan lebih mudah pula dalam melakukan tahapan selanjutnya. *Fully developed use case description* juga diperlukan untuk meningkatkan pemahaman mengenai proses bisnis. Pada tahap ini juga didefinisikan *output* yang akan dihasilkan, fitur-fitur yang dimiliki aplikasi dan fungsi aplikasi yang dikembangkan, seperti fitur cek harga, *post* harga, dan aspirasi.

### Desain (XP Design)

Tahap desain dilakukan dengan mengidentifikasi dan mengorganisasikan kelas-kelas yang digunakan dalam pembangunan aplikasi. Pada tahap ini dibuat *mockup* sebagai acuan kerja dalam pembuatan *website* agar tidak menyimpang dari tujuan awal pembuatannya. Pada saat pembuatan *mockup*, kerangka pembangunan *website* sudah ditentukan terlebih dahulu sehingga eksekusi pembangunan *website* menjadi lebih efektif dan tersruktur. Pada tahap ini juga dibuat *class diagram* untuk memperlihatkan antar kelas dan penjelasan detail tiap kelas di dalam model desain sistem yang dibangun. Tahap desain dilakukan sebelum dan setelah tahap pengkodean, sehingga tahapan desain dilakukan terus-menerus selama proses pembangunan aplikasi berlangsung.

### Pengkodean (XP Coding)

Tahap pengkodean diawali dengan membangun unit pengujian sebelum melakukan pengkodean secara keseluruhan. XP juga lebih menekankan *pair programming* yang melibatkan lebih dari satu orang untuk melakukan pengkodean, untuk kemudian digabungkan menjadi satu sistem yang utuh. Pengkodean dilakukan menggunakan bahasa pemrograman PHP dan Node.js dengan *framework* Code Igniter 3.03.

### Pengujian (XP Testing)

Tahap ini merupakan tahap terakhir dari metode XP. Pengujian dilakukan dengan menguji fitur dan fungsionalitas aplikasi yang telah dibangun. Pada tahapan ini aplikasi diujikan sebanyak dua kali dengan metode pengujian *black-box* oleh tim secara internal.

### Perangkat Penelitian

Spesifikasi perangkat keras dan perangkat lunak yang digunakan untuk penelitian ini yaitu:

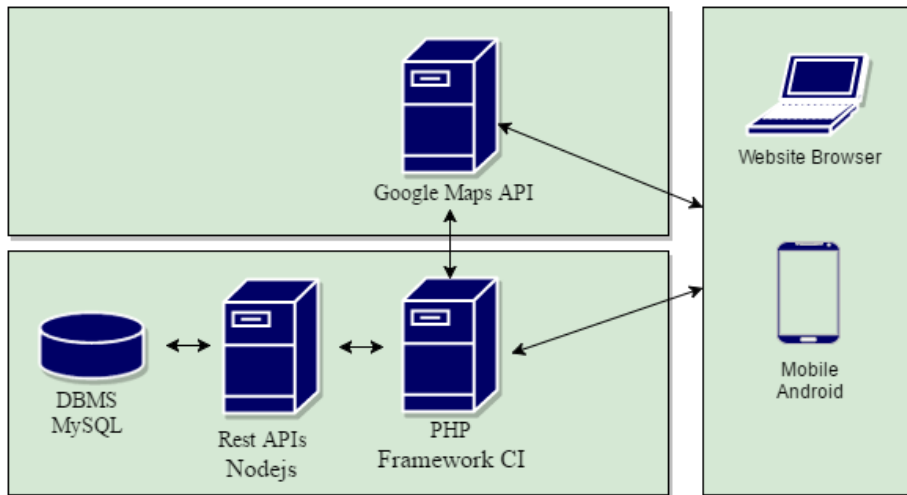
- Perangkat keras:
  - *Harddisk* 1 TB
  - RAM 4 GB
  - *Processor* Intel Core i5 @2.50 GHz
- Perangkat lunak:
  - Sistem operasi Windows 10
  - Sublime Text sebagai *text editor*
  - Apache2 sebagai *web server*
  - Bahasa pemrograman Node.js dan PHP dengan *framework* Code Igniter 3.03
  - DBMS MySQL
  - Postman sebagai *software* penguji REST APIs
  - Balsamiq sebagai *software* pembuatan *mockup*

## HASIL DAN PEMBAHASAN

### Iterasi Pertama

#### Perencanaan (XP Planning)

Tahap perencanaan pada iterasi pertama dimulai dengan membangun arsitektur penelitian yang dapat dilihat pada Gambar 2. Basis data yang digunakan yaitu MySQL. REST API yang dibangun menggunakan bahasa pemrograman Node.js digunakan sebagai perantara *request* dari klien ke *server*, sehingga *client*, dalam hal ini pengguna web dan *mobile* lebih mudah dan cepat dalam mengonsumsi data. Bahasa pemrograman PHP dengan *framework* Code Igniter (CI) 3.03 digunakan untuk membuat tampilan pada halaman web. *Google Maps API* digunakan untuk menampilkan peta dan menandai lokasi yang didatangi masyarakat dengan mengambil *longitude* dan *latitude*.



Gambar 2 Arsitektur penelitian aplikasi Portal Harga

Pada tahap ini juga dibuat *user story*, *use case diagram*, *fully developed use case description*, dan *swimlane diagram*. Tabel 1 menunjukkan *user story* aplikasi Portal Harga dari sisi pengguna masyarakat. Pada *user story* terdapat prioritas yang diurutkan berdasarkan tingkat kepentingan *task* yang harus dibuat.

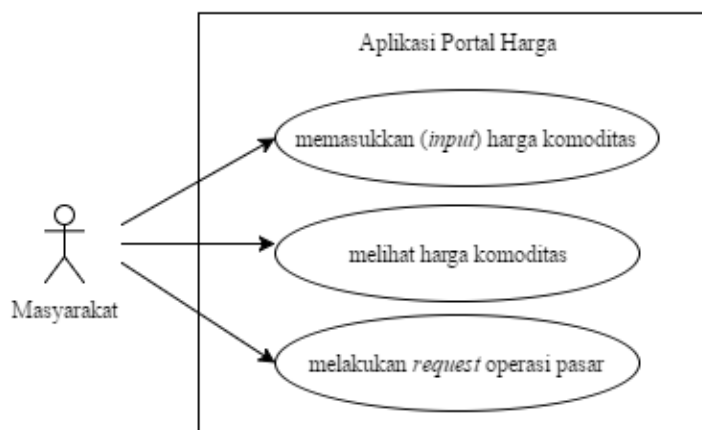
Tabel 1 *User story* Portal Harga

Actor	Task	User story	Priority
Masyarakat	Memasukkan (input) harga komoditas (post harga)	Masyarakat ingin memasukkan data harga dan lokasi tempat ia membeli komoditas (misalnya beras di pasar A) sehingga ia bisa menyebarkan informasi kepada masyarakat lain terkait harga komoditas saat ini.	1
	Mengetahui harga komoditas (cek harga)	Masyarakat ingin mengetahui harga komoditas di pasar, sehingga ia bisa mendapatkan informasi harga yang sesuai dengan kebutuhannya.	2



Melakukan <i>request</i> operasi pasar (aspirasi)	Masyarakat yang mengalami gejala peningkatan harga yang cukup tinggi di pasar ingin melapor kepada pemerintah terkait lonjakan harga tersebut sehingga pemerintah dapat menanggulangnya secara cepat dengan melakukan operasi pasar.	3
---	--	---

Pada *use case diagram*, terlihat bahwa masyarakat dapat memasukkan harga komoditas, melihat harga komoditas, dan melakukan *request* operasi pasar. Adapun *use case diagram* Portal Harga dapat dilihat pada Gambar 3. *Fully developed use case description* untuk fungsi memasukkan (*input*) harga komoditas (*post* harga) ditunjukkan oleh Tabel 2. *Fully developed use case description* untuk fungsi melihat harga komoditas (cek harga) dan *request* operasi pasar (aspirasi) dapat dilihat pada Lampiran 1 dan 2.



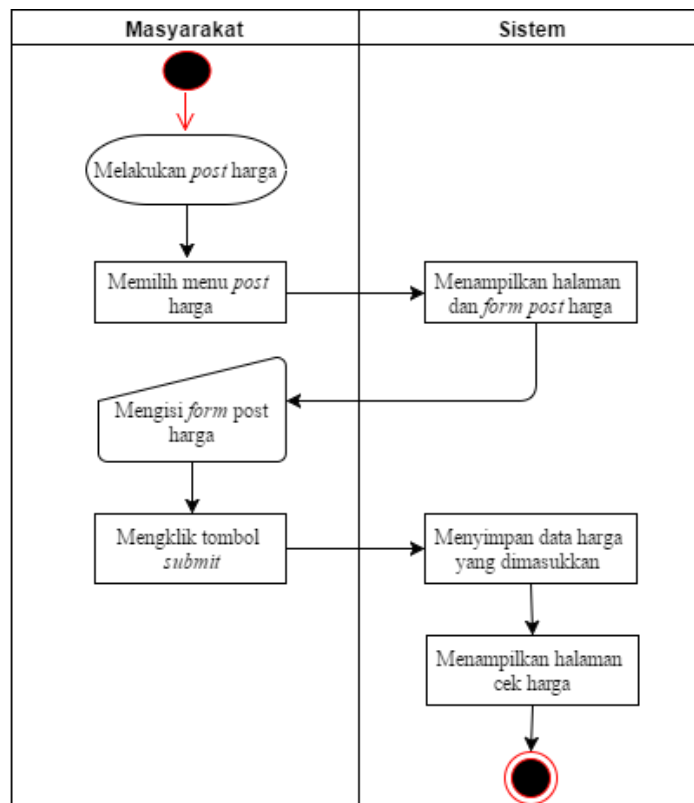
Gambar 3 *Use case diagram* Portal Harga dari sisi pengguna masyarakat

Tabel 2 *Fully developed use case description* fungsi *post* harga

<i>Use case Name:</i>	Memasukkan ( <i>input</i> ) harga komoditas ( <i>post</i> harga)
<i>Scenario:</i>	Masyarakat ingin memasukkan harga komoditas pertanian
<i>Triggering Event:</i>	Masyarakat memilih menu <i>Post</i> Harga
<i>Brief Description:</i>	Masyarakat memasukkan data harga dan lokasi komoditas pertanian
<i>Actors:</i>	Masyarakat
<i>Related Use Case:</i>	-
<i>Stakeholders:</i>	Masyarakat
<i>Preconditions:</i>	Masyarakat telah melakukan <i>login</i> dan memilih menu <i>post</i> harga

<i>Postconditions:</i>	Masyarakat berhasil memasukkan data harga komoditas	
<i>Flow of Activities:</i>	<i>Actor</i>	<i>System</i>
	1 Memilih menu post harga	1.1 Menampilkan halaman <i>post</i> harga
	2 Mencari lokasi pembelian komoditas	
	3 Mengisi <i>form</i> post harga	
	4 Mengklik tombol post harga	4.1 Menyimpan data <i>input</i> dan menampilkannya pada halaman cek harga

Setelah membuat *use case diagram*, *user story* dan *fully developed use case description*, selanjutnya dibuat *swimlane diagram* untuk menunjukkan *stakeholder* yang bertanggung jawab dalam aktivitas bisnis pelaporan harga. Gambar 4 menunjukkan *swimlane diagram* untuk fungsi *post* harga, sedangkan *swimlane diagram* untuk fungsi cek harga dapat dilihat pada Lampiran 3.

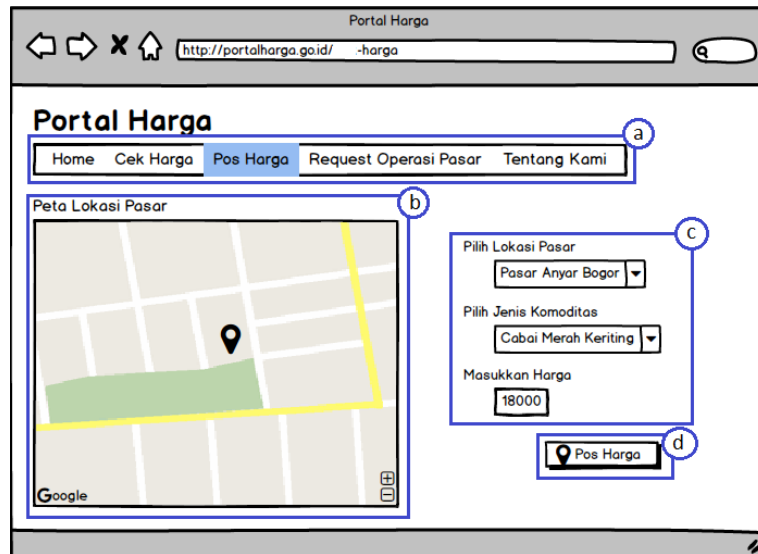


Gambar 4 *Swimlane diagram* fungsi *post* harga

### Desain (XP Design)

Tahap desain dilakukan dengan membuat *mockup* untuk keseluruhan fungsi aplikasi, yaitu fungsi *post* harga, cek harga, dan aspirasi. Gambar 5 menampilkan *mockup* untuk fungsi *post* harga, sedangkan *mockup* untuk fungsi cek harga dan aspirasi dapat dilihat pada Lampiran 4 dan 5. Pada halaman *post* harga, pengguna memilih lokasi pasar dan jenis komoditas. Setelah memilih pasar dan jenis komoditas, pengguna memasukkan harga pada kolom harga yang tersedia.

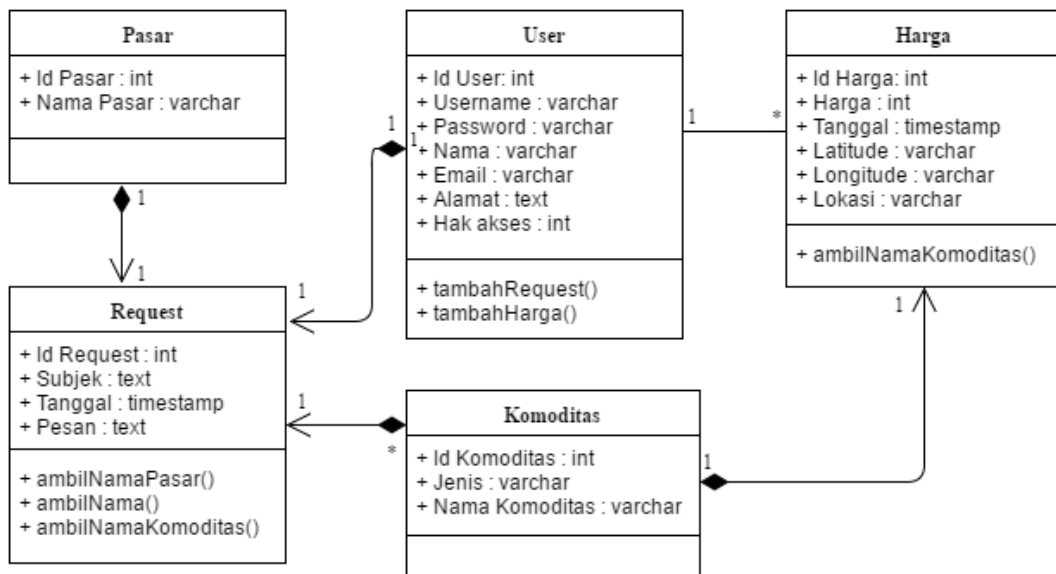
Pengguna mengklik tombol *post* harga kemudian sistem menyimpan data yang telah dimasukkan pengguna.



Gambar 5 Mockup halaman *post* harga

Pada Gambar 5, terdapat bagian-bagian halaman *post* harga yang dibatasi oleh kotak berwarna biru dengan masing-masing diberi huruf a sampai d. Bagian (a) berisi menu navigasi yang terdiri dari *home*, *cek harga*, *post harga*, *request operasi pasar*, dan *tentang kami*. Pada *mockup*, terlihat menu yang aktif yaitu menu *post harga*. Bagian (b) menunjukkan peta lokasi pasar, bagian (c) berisi pilihan lokasi pasar dan jenis komoditas, serta kolom harga yang harus diisi pengguna. Bagian (d) merupakan tombol *submit post* harga setelah semua data dimasukkan.

Pada tahap ini juga *class diagram* yang terdiri dari kelas *user*, komoditas, pasar, harga, dan *request*. Adapun *class diagram* aplikasi portal harga ditunjukkan oleh Gambar 6.



Gambar 6 Basis data aplikasi Portal Harga

### Pengkodean (XP Coding)

Pada tahap ini dilakukan *pair programming* secara internal oleh tim yang bertanggungjawab dalam pembangunan aplikasi web Portal Harga keseluruhan, meliputi modul pengguna masyarakat, petani, pemerintah, dan admin. Tahap pengkodean dilakukan dengan membangun REST APIs untuk fungsi *post* harga dan cek harga menggunakan *method* POST dan GET. Sebelum dibangun, REST harus terlebih dahulu dihubungkan dengan basis data. Gambar 7 menunjukkan kode untuk menghubungkan REST dengan basis data yang telah dibuat.

```

1  REST.prototype.connectMysql = function() {
2      var self = this;
3      var pool      =      mysql.createPool({
4          connectionLimit : 100,
5          host           : 'localhost',
6          user           : 'root',
7          password       : '',
8          database       : 'portal',
9          debug          : false
10     });
11     pool.getConnection(function(err,connection) {
12         if(err) {
13             self.stop(err);
14         } else {
15             self.configureExpress(connection);
16         }
17     });
18 }
19
20 REST.prototype.configureExpress = function(connection) {
21     var self = this;
22     app.use(bodyParser.urlencoded({ extended: true }));
23     app.use(bodyParser.json());
24     var router = express.Router();
25     app.use('/api_portalharga', router);
26     var rest_router = new rest(router,connection,md5);
27     self.startServer();
28 }
29
30 REST.prototype.startServer = function() {
31     app.listen(3000,function() {
32         console.log("All right ! I am alive at Port 3000.");
33     });
34 }

```

Gambar 7 Kode program untuk menghubungkan REST dengan basis data

Pada potongan kode, dapat dilihat bahwa REST dihubungkan dengan *localhost* sebagai *host* dan basis data yang digunakan yaitu *portal*. *Base route* yang digunakan yaitu *api\_portalharga* yang dijalankan pada *port* 3000. Ketika *server* portal harga berhasil dijalankan, muncul pesan yaitu “All right ! I am alive at Port 3000.”

Tahap selanjutnya yaitu membangun REST API untuk fungsi *post* harga dan cek harga. Fungsi REST untuk *post* harga dibangun menggunakan *method* POST dengan bahasa pemrograman Node.js. Kode program pembuatan REST untuk fungsi *post* harga dapat dilihat pada Gambar 8.

```

497 | router.post("/post_harga",function(req, res){
498 |     var query = "INSERT INTO ??(?,?,?,?,,?,?)
VALUES (?,?,?,?,?,?);";
499 |     var table =
["harga","harga","tanggal","latitude","longitude","lokasi",
"id_komoditas","id_user",req.body.harga,req.body.tanggal,req
body.latitude,req.body.longitude,req.body.lokasi,req.body
.id_komoditas,req.body.id_user];
500 |     query = mysql.format(query,table);
501 |     connection.query(query,function(err,rows){
502 |         if(err) {
503 |             res.json({"Error" : true, "Message" : "Error
executing MySQL query"});
504 |         } else {
505 |             res.json({"Error" : false, "Message" : "Post
harga berhasil!" });
506 |         }
507 |     });
508 | });

```

Gambar 8 Kode program REST API untuk fungsi *post* harga

Pada Gambar 8, terlihat bahwa REST dibangun menggunakan *method* POST yang ditunjukkan oleh `router.post`. REST untuk fungsi *post* harga dapat dijalankan dengan mengakses url `http://localhost:3000/api_portalharga/post_harga`. *Method* POST memasukkan *record* ke dalam tabel harga dengan mengisi beberapa *field*, yaitu harga, tanggal, *latitude*, *longitude*, lokasi, *id\_komoditas*, dan *id\_user*. Jika *post* harga gagal, muncul pesan kesalahan seperti pada baris 503. Ketika *post* harga berhasil dilakukan, muncul pesan seperti pada baris 505, yaitu "Error" : false, "Message" : Post harga berhasil!" yang menunjukkan bahwa *record* berhasil ditambahkan ke dalam tabel harga.

Fungsi REST cek harga dibangun menggunakan *method* GET yang kodenya dapat dilihat pada Gambar 9. *Method* GET ditunjukkan oleh `router.get` pada baris 86. Pada kode terlihat bahwa data diambil dari tabel harga dan komoditas. Sama seperti pada *method* POST, pesan kesalahan akan muncul ketika cek harga gagal dilakukan. Jika cek harga berhasil, muncul data dengan format teks JSON yang dapat dilihat pada Gambar 10.

```

86 | router.get("/cek_harga",function(req,res){
87 |     var query ="SELECT * FROM ??,?? where
harga.id_komoditas=komoditas.id_komoditas";
88 |     var table=["harga","komoditas"];
89 |     query=mysql.format(query,table);
90 |     connection.query(query,function(err,data){
91 |         if(err) {
92 |             res.json({"Error" : true, "Message" : "Error
executing MySQL query"});
93 |         } else {
94 |             res.json({data});
95 |         }
96 |     });
97 | });

```

Gambar 9 Kode program REST API untuk fungsi cek harga

```

1  {
2    "data": [
3      {
4        "id_harga": 4,
5        "harga": 15000,
6        "tanggal": "2016-06-09",
7        "latitude": "-6.1748556000000001",
8        "longitude": "106.845035100000002",
9        "lokasi": "Jalan Bungur Besar No. 169, Bungur, Senen,
        Bungur, Jakarta Pusat, Kota Jakarta Pusat, Daerah Khusus
        Ibukota Jakarta, Indonesia",
10       "id_komoditas": 1,
11       "id_user": 5,
12       "jenis": "Beras",
13       "nama_komoditas": "Beras "
14     }
15   ]
16 }

```

Gambar 10 Data teks JSON hasil dari fungsi cek harga

Data teks JSON dari fungsi cek harga pada Gambar 10 didapatkan dengan mengakses url [http://localhost:3000/api\\_portalharga/cek\\_harga](http://localhost:3000/api_portalharga/cek_harga). Array JSON ditunjukkan oleh *data* yang terdiri dari satu objek. Objek berada dalam kurung siku setelah nama *array* dan dipisahkan dengan kurung kurawal antara objek yang satu dengan lainnya. Objek pada fungsi cek harga memiliki *field* *id\_harga*, *harga*, *tanggal*, *latitude*, *longitude*, *lokasi*, *id\_komoditas*, *id\_user*, *jenis*, dan *nama\_komoditas*. *Id\_harga* merupakan urutan data harga yang masuk ke basis data. *Harga* merupakan data harga komoditas yang dimasukkan oleh pengguna. *Tanggal* menunjukkan waktu ketika pengguna memasukkan harga. *Latitude*, *longitude* dan *lokasi* didapatkan dari Google Maps API saat pengguna melakukan pencarian lokasi pada peta. *Id\_komoditas* merupakan nomor identitas untuk menandai komoditas tertentu, dalam hal ini *id\_komoditas* 1 merupakan komoditas dengan jenis dan nama beras. *Id\_user* merupakan nomor identitas bagi pengguna yang telah melakukan registrasi.

### Pengujian (*Testing*)

Tahap pengujian dilakukan menggunakan *software* Postman. Fungsi REST yang diujikan yaitu POST dan GET dari fungsi *post* harga dan cek harga. Kedua fungsi REST yang dibuat telah berhasil dijalankan dengan baik. Hasil pengujian fungsi REST dapat dilihat pada Tabel 3.

Tabel 3 Pengujian fungsi REST iterasi pertama

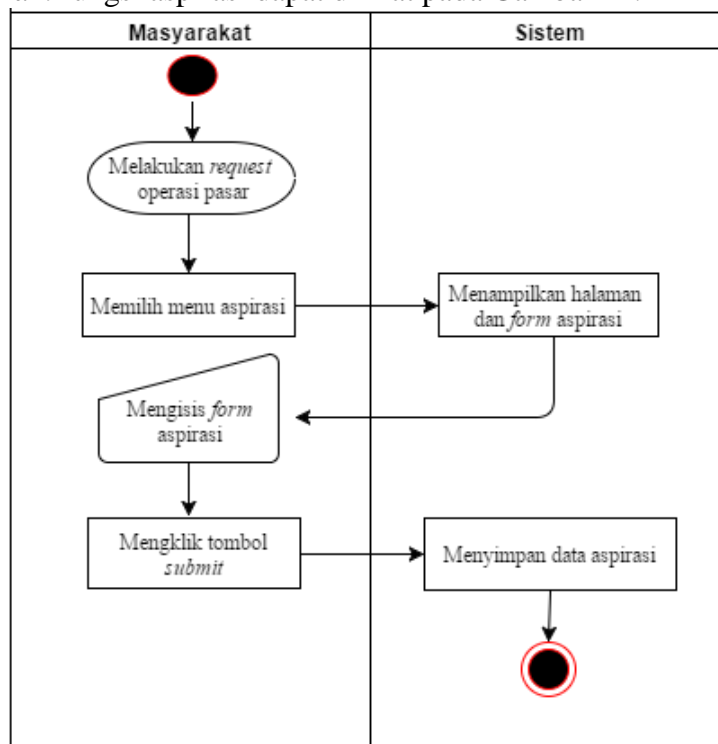
Aktor	Fungsi REST	Status Pengujian
Masyarakat	<i>POST</i> Post Harga	Berhasil
	<i>GET</i> Cek Harga	Berhasil

### Iterasi Kedua

#### Perencanaan (*XP Planning*)

Pada tahap perencanaan iterasi kedua dibangun *swimlane diagram* untuk fungsi aspirasi yang belum dibuat pada iterasi pertama. Seperti pada iterasi

pertama, tujuan dibangunnya *swimlane diagram* yaitu untuk menunjukkan aktivitas yang dilakukan pengguna dalam melakukan *request* operasi pasar. *Swimlane diagram* fungsi aspirasi dapat dilihat pada Gambar 11.



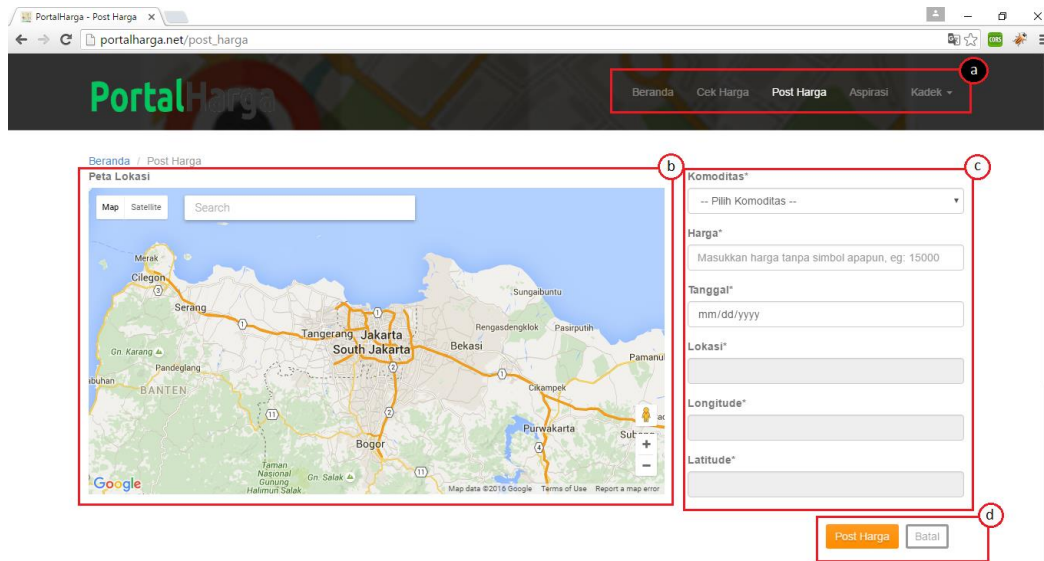
Gambar 11 *Swimlane diagram* fungsi aspirasi

### Desain (XP Design)

Tahap desain dilakukan dengan memperbaharui tampilan fungsi *post* harga dengan menambahkan fitur pencarian lokasi yang terhubung dengan *Google Maps* langsung pada tampilan web. Pada fungsi cek harga juga dilakukan perubahan, yaitu menghapus fitur pencarian jenis komoditas dan lokasi, sehingga pengguna hanya dapat mengetahui harga komoditas dengan mengklik *marker* pada peta yang ditampilkan. Tampilan halaman fungsi aspirasi juga diubah dengan menghilangkan peta lokasi, sehingga menyisakan *field-field* yang harus diisi oleh pengguna. Gambar 12 menunjukkan tampilan fungsi *post* harga pada halaman web.

Pada Gambar 12, dapat dilihat bahwa halaman dibagi menjadi empat bagian yang dibatasi oleh kotak berwarna merah yang masing-masing berisi keterangan huruf a sampai d. Bagian (a) merupakan menu navigasi yang terdiri dari beranda, cek harga, *post* harga, aspirasi, dan profil. Pada gambar terlihat menu navigasi yang aktif yaitu menu *post* harga. Bagian (b) menunjukkan peta lokasi dengan kolom pencarian lokasi yang memudahkan pengguna mencari lokasi komoditas. Bagian (c) merupakan *form* yang berisi *field-field* yang harus diisi, yaitu memilih komoditas, mengisi kolom harga, memilih tanggal pelaporan harga, lokasi, *longitude* dan *latitude*. Lokasi, *longitude*, dan *latitude* pada *form* ini dibuat *fix* dan otomatis terisi setelah pengguna memilih lokasi pada kolom pencarian untuk menghindari perubahan lokasi yang dipilih. Bagian (d) merupakan tombol *submit* *post* harga dan batal ketika pengguna telah mengisi seluruh *field* yang tersedia.

Tampilan halaman untuk menu beranda, cek harga, dan aspirasi dapat dilihat pada Lampiran 6 sampai Lampiran 8.



Gambar 12 Tampilan halaman post harga pada web

### Pengkodean (XP Coding)

Pada tahap pengkodean dilakukan pembangunan REST API untuk fungsi aspirasi menggunakan *method* POST. Seperti pada iterasi pertama, sebelum dibangun, REST harus terlebih dahulu dihubungkan dengan basis data. Pada tahap ini juga dilakukan penambahan token OAuth untuk menghindari pihak yang tidak berwenang dalam mengakses APIs. Token yang digunakan masih bersifat statis, sehingga harus selalu ditambahkan pada url yang digunakan untuk mengakses REST. Gambar 13 menunjukkan potongan kode untuk menambahkan token pada aplikasi portal harga.

```

42 | router.use(function(req,res,next){
43 |     var token = req.body.token || req.query.token ||
      req.headers['x-access-token'];
44 |     if(token == 'portalharga'){
45 |         next();
46 |     }else{
47 |         return res.status(403).send({
48 |             "Error": true,
49 |             "Message": 'No token provided.'
50 |         });
51 |     }
52 | });

```

Gambar 13 Kode program penambahan token OAuth

Pembangunan REST untuk fungsi aspirasi menggunakan *method* POST yang potongan kodenya dapat dilihat pada Gambar 14. Hal ini ditunjukkan oleh `router.post`. Pada gambar, dapat dilihat bahwa fungsi aspirasi, dalam hal ini *request* berada pada *path* `/request`, sehingga fungsi REST ini harus diakses melalui `http://localhost:3000/api_portalharga/request?token=portalharga`. Portal harga di akhir url merupakan token yang telah dibuat dan harus selalu ditambahkan untuk mengakses REST, termasuk fungsi REST yang sebelumnya



telah dibuat pada iterasi pertama. Pada pembangunan REST untuk fungsi aspirasi, *record* dimasukkan ke dalam tabel *request*, dengan mengisi enam *field*, yaitu *id\_user*, subjek, tanggal, *id\_komoditas*, *id\_pasar*, dan pesan. Jika penambahan aspirasi gagal, muncul pesan kesalahan seperti pada baris 470, sedangkan jika berhasil, data aspirasi berhasil ditambahkan dan muncul pesan seperti pada baris 472, yaitu “Error” : false, “Message” : “Request Added !”.

```

464 | router.post("/request", function(req,res){
465 |     var query="INSERT INTO ??(?,?,?,?,,?) VALUES
      |     (?,,?,,?,?) ";
466 |     var
      |     table=["request","id_user","subjek","tanggal","id_komod
      |     itas","id_pasar","pesan",req.body.id_user,req.body.subj
      |     ek,req.body.tanggal,req.body.id_komoditas,req.body.id_p
      |     asar,req.body.pesan];
467 |     query = mysql.format(query,table);
468 |     connection.query(query,function(err,rows){
469 |         if(err){
470 |             res.json({"Error" : true, "Message": "Error executing
      |             MySQL query"});
471 |             }else{
472 |                 res.json({"Error" : false, "Message" : "Request Adedd
      |                 !"});
473 |             }
474 |         });
475 |     });

```

Gambar 14 Kode program REST API untuk fungsi aspirasi

Pada tahap ini juga dilakukan pengimplementasian REST APIs ke dalam web yang telah dibangun menggunakan *framework* Code Igniter 3.03. Salah satu contoh kode *script* implementasi untuk fungsi *post* harga dapat dilihat pada Gambar 15. Kode *script* ini digunakan untuk memanggil url dari REST untuk kemudian ditampilkan ke dalam web menggunakan bahasa pemrograman AJAX. *Method* yang digunakan yaitu POST dan mengambil *id #form-add* yang disesuaikan dengan *id* pada kode html.

```

75 | function add_post_harga(){
76 |     $.ajax({
77 |         type: 'POST',
      |         url:
      |         "http://localhost:3000/api_portalharga/post_harga?toke
      |         n=portalharga",
79 |         data: $("#form-add").serialize(),
80 |         success: function(msg) {
81 |             window.location = "cek_harga";
82 |             console.log(msg);
83 |         }
84 |     });

```

Gambar 15 Kode *script* implementasi fungsi *post* harga

Setelah dihubungkan, dilakukan implementasi REST dengan tampilan web yang dibuat menggunakan bahasa pemrograman PHP, yang bertujuan untuk memudahkan pengguna dalam melihat dan memahami data teks JSON yang dihasilkan oleh REST. Potongan kode html untuk tampilan halaman *post* harga dapat dilihat pada Lampiran 9. *Field-field* yang harus diisi pengguna masuk ke dalam *form-group*, sedangkan *script* googleapis digunakan untuk menampilkan

peta. Tampilan halaman aplikasi web hasil implementasi sama dengan tampilan pada tahap desain iterasi kedua. Seperti pada iterasi pertama, pada tahap ini dilakukan *pair programming* secara internal oleh tim yang bertanggungjawab dalam pembangunan aplikasi web Portal Harga keseluruhan, meliputi modul pengguna masyarakat, petani, pemerintah, dan admin.

### Pengujian (*Testing*)

Tahap pengujian iterasi kedua dilakukan dengan mengujikan REST APIs yang telah dibuat beserta implementasi REST APIs ke dalam aplikasi web. Pengujian dilakukan oleh tim secara internal menggunakan *software* Postman. Hasil pengujian REST APIs dapat dilihat pada Tabel 4.

Tabel 4 Pengujian fungsi REST iterasi kedua

Aktor	Fungsi REST	Status Pengujian
Masyarakat	<i>POST Post Harga</i>	Berhasil
	<i>GET Cek Harga</i>	Berhasil
	<i>POST Aspirasi</i>	Berhasil
	<i>GET Login</i>	Berhasil
	<i>POST Register</i>	Berhasil
	<i>GET Logout</i>	Berhasil

## SIMPULAN DAN SARAN

### Simpulan

Penelitian ini berhasil mengembangkan aplikasi web Portal Harga berbasis REST APIs. Aplikasi web ini dapat digunakan masyarakat dalam mengetahui harga komoditas sehingga masyarakat dapat memilih harga yang sesuai dengan kebutuhan. Aplikasi web ini juga dapat memfasilitasi masyarakat dalam melakukan pelaporan harga dan memantau perkembangan harga, serta menyampaikan aspirasi masyarakat terkait ketidakstabilan harga di pasar.

### Saran

Rest APIs yang dibangun sebaiknya menggunakan struktur NoSQL untuk meningkatkan performa *database*. Perlu dibuat fungsi *filter* pada menu cek harga untuk mengoptimalkan penggunaan web. Penambahan keamanan untuk memvalidasi informasi yang dimasukkan oleh pengguna juga diperlukan, salah satunya dengan membuat fungsi *login* menggunakan akun media sosial.

## DAFTAR PUSTAKA

- [APJII] Asosiasi Penyelenggara Jasa Internet Indonesia. 2014. Profil pengguna internet Indonesia [Internet]. [diunduh 2015 Des 6]. Tersedia pada: <http://www.slideshare.net/internetsehat/profil-pengguna-internet-indonesia-2014-riset-oleh-apjii-dan-puskakom-ui>.

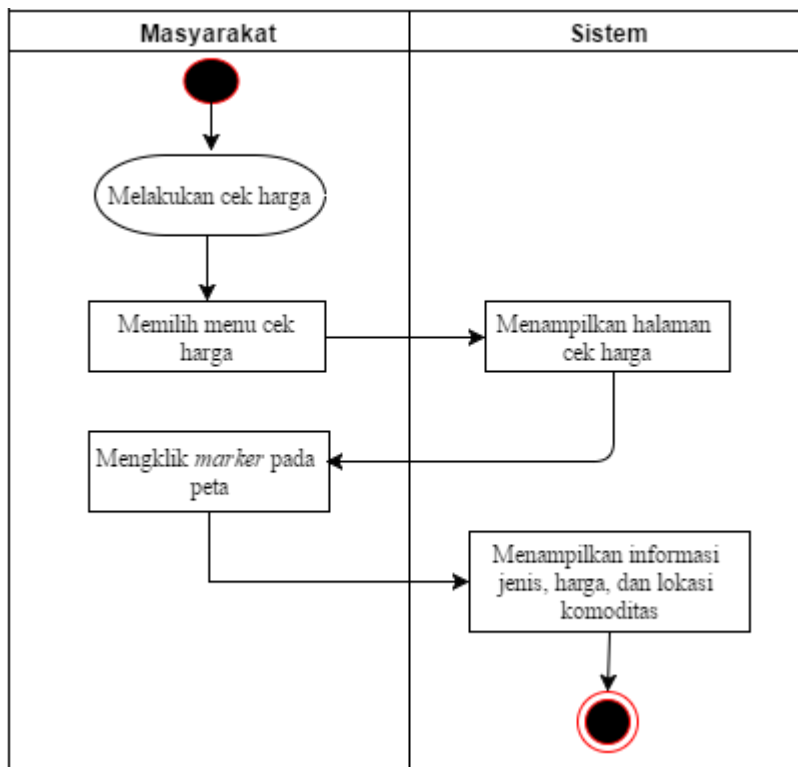
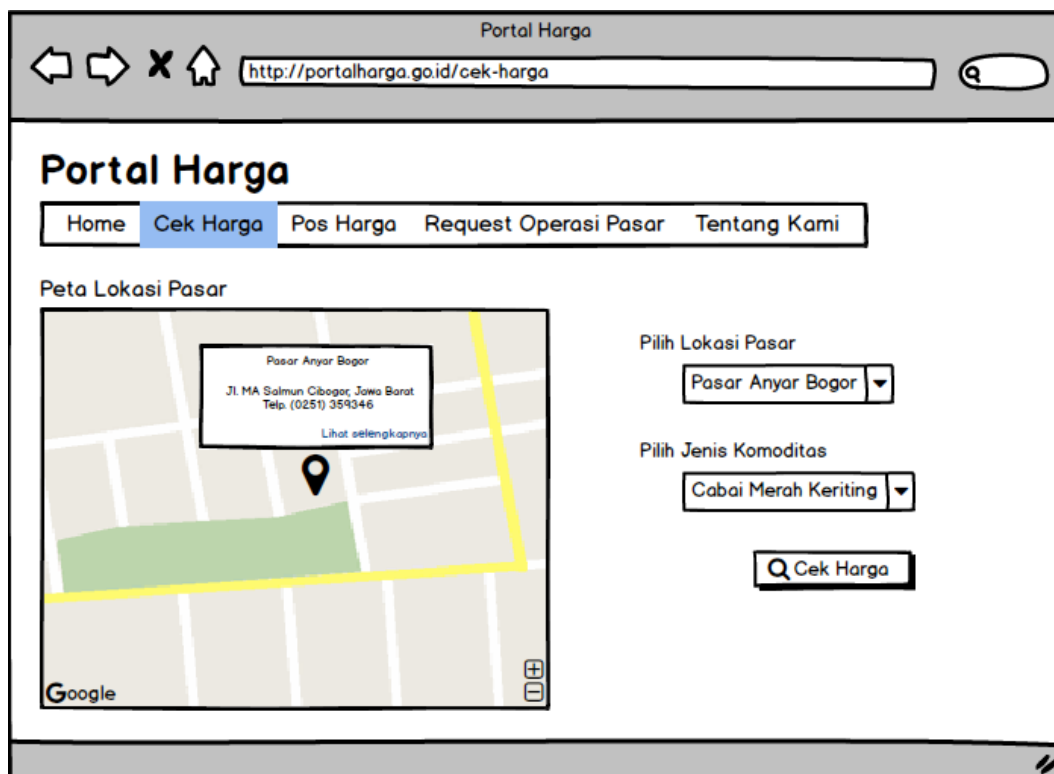
- Astika IW, Solahudin M, Maradona RF, Nugraha MPA. 2009. Sistem informasi harga produk pertanian berbasis SMS [Internet]. [diunduh 2015 Nov 29]. Tersedia pada: <http://repository.ipb.ac.id/handle/123456789/31437>.
- Fauziah Y. 2013. Aplikasi iklan baris online menggunakan arsitektur REST web service [Internet]. *Telematika*. 9(2):75-80. [diunduh 2016 Jan 10]. Tersedia pada: <http://jurnal.upnyk.ac.id/index.php/telematika/article/download/286/248>.
- Istiyanto JE, Sutanta E. 2012. Model interoperabilitas antar aplikasi e-government. *Technoscientia* [Internet]. [diunduh 2016 Jan 10]; 4(2). Tersedia pada: [https://www.researchgate.net/publication/235525921\\_Model\\_Interoperabilitas\\_Antar\\_Aplikasi\\_e-Government](https://www.researchgate.net/publication/235525921_Model_Interoperabilitas_Antar_Aplikasi_e-Government).
- Pemerintah Republik Indonesia. 2013. Undang-Undang Republik Indonesia Nomor 19 Tahun 2013 tentang Perlindungan dan Pemberdayaan Petani. Jakarta (ID): Sekretariat Negara.
- Pressman RS. 2010. *Software Engineering: A Practitioner's Approach*. Ed ke-7. New York(US): McGraw-Hill.
- Widodo. 2008. Extreme Programming: Pengembangan perangkat lunak semi formal. [Internet]. [diunduh 2015 Des 3]. Tersedia pada: [http://iatt.kemenperin.go.id/tik/fullpaper/fullpaper15\\_widodo.pdf](http://iatt.kemenperin.go.id/tik/fullpaper/fullpaper15_widodo.pdf).

Lampiran 1 *Fully developed use case description* fungsi melihat harga komoditas (cek harga)

<i>Use case Name:</i>	Melihat harga komoditas (cek harga)	
<i>Scenario:</i>	Masyarakat ingin melihat harga komoditas pertanian	
<i>Triggering Event:</i>	Masyarakat memilih menu Cek Harga	
<i>Brief Description:</i>	Masyarakat memilih lokasi komoditas dan melihat harganya	
<i>Actors:</i>	Masyarakat	
<i>Related Use Case:</i>	-	
<i>Stakeholders:</i>	Masyarakat	
<i>Preconditions:</i>	Masyarakat telah melakukan <i>login</i> dan memilih menu Cek Harga	
<i>Postconditions:</i>	Masyarakat berhasil melihat data harga komoditas	
<i>Flow of Activities:</i>	<i>Actor</i>	<i>System</i>
	1 Memilih menu Cek Harga	1.1 Menampilkan halaman Cek Harga
	2 Mengklik <i>marker</i> pada peta di halaman Cek Harga	2.1 Menampilkan harga komoditas

Lampiran 2 *Fully developed use case description* fungsi melakukan *request* operasi pasar (aspirasi)

<i>Use case Name:</i>	Melakukan request operasi pasar (aspirasi)						
<i>Scenario:</i>	Masyarakat ingin melakukan <i>request</i> operasi pasar kepada pemerintah						
<i>Triggering Event:</i>	Masyarakat memilih menu Aspirasi						
<i>Brief Description:</i>	Masyarakat memasukkan pesan dan memilih jenis komoditas dan lokasi pasar						
<i>Actors:</i>	Masyarakat						
<i>Related Use Case:</i>	-						
<i>Stakeholders:</i>	Masyarakat						
<i>Preconditions:</i>	Masyarakat telah melakukan <i>login</i> dan memilih menu Aspirasi						
<i>Postconditions:</i>	Masyarakat berhasil melakukan <i>request</i> operasi pasar						
<i>Flow of Activities:</i>	<table> <tr> <th><i>Actor</i></th><th><i>System</i></th></tr> <tr> <td>1 Memilih menu Aspirasi</td><td>1.1 Menampilkan halaman Aspirasi</td></tr> <tr> <td>2 Mengisi form <i>request</i> operasi pasar</td><td>2.1 Menyimpan data input</td></tr> </table>	<i>Actor</i>	<i>System</i>	1 Memilih menu Aspirasi	1.1 Menampilkan halaman Aspirasi	2 Mengisi form <i>request</i> operasi pasar	2.1 Menyimpan data input
<i>Actor</i>	<i>System</i>						
1 Memilih menu Aspirasi	1.1 Menampilkan halaman Aspirasi						
2 Mengisi form <i>request</i> operasi pasar	2.1 Menyimpan data input						

Lampiran 3 *Swimlane diagram fungsi cek harga*Lampiran 4 *Mockup fungsi cek harga*

Lampiran 5 Mockup fungsi aspirasi

Portal Harga

Home Cek Harga Pos Harga **Request Operasi Pasar** Tentang Kami

Peta Lokasi Pasar

Pasar Anyar Bogor  
Jl. MA Salmun Cibogor, Jawa Barat  
Telp. (0251) 359346  
[Lihat selengkapnya](#)

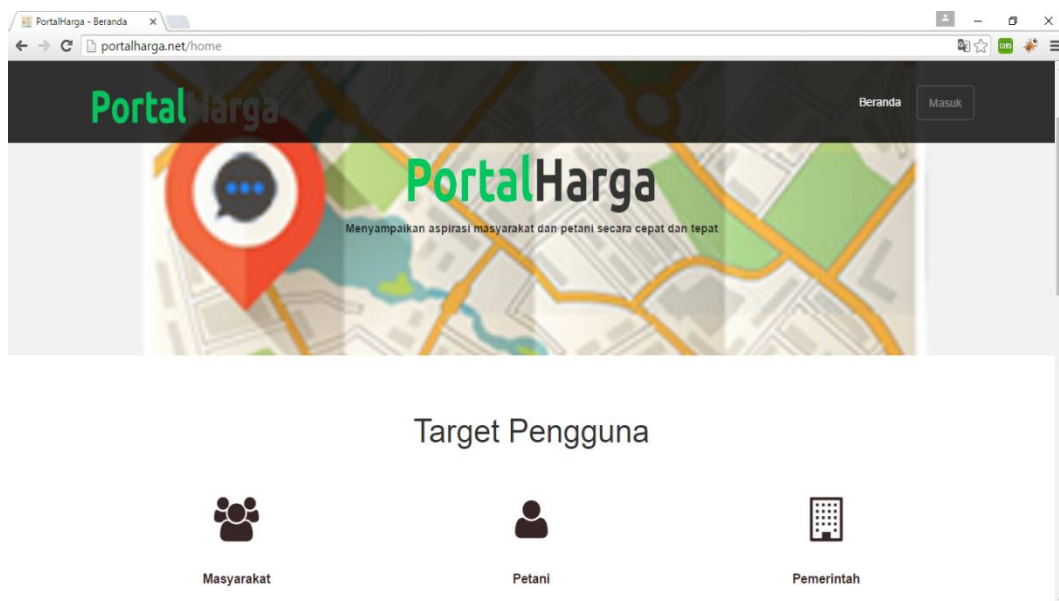
Pilih Lokasi Pasar  
Pasar Anyar Bogor

Laporan

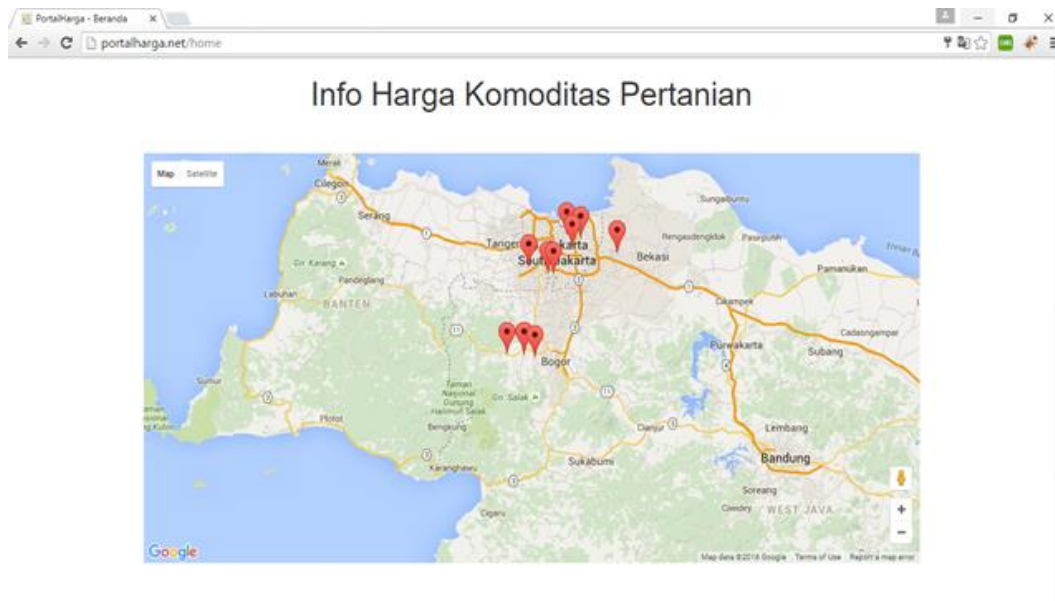
Di pasar Anyar terjadi kenaikan harga cabai rawit merah yang semula 18000 menjadi 32000. Tolong segera ditanggulangi ya, Pak. Terima kasih.

Kirim Request

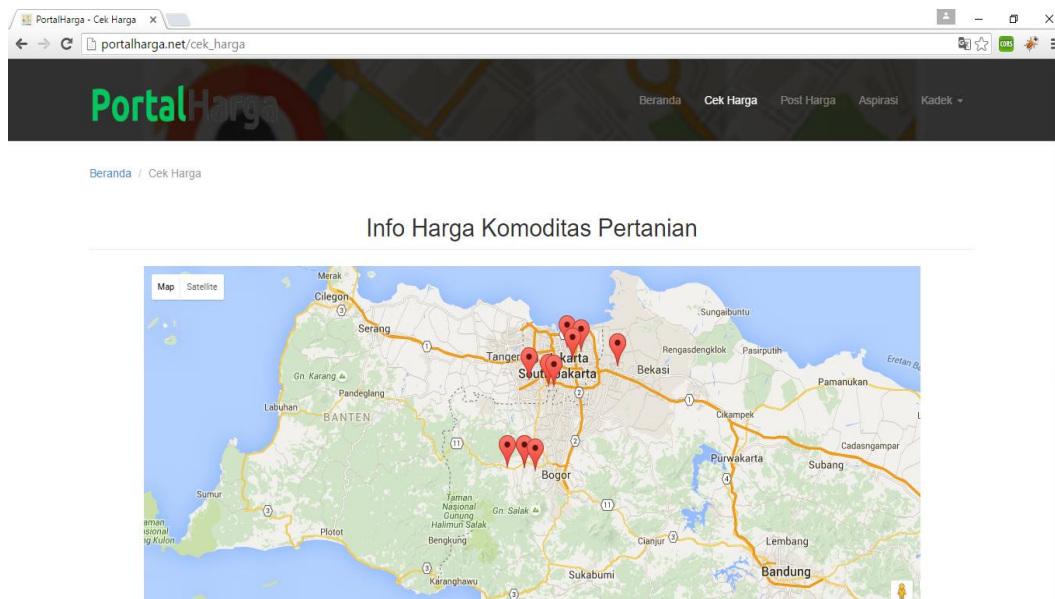
Lampiran 6 Tampilan halaman beranda



## Lampiran 7 Lanjutan



## Lampiran 8 Tampilan halaman cek harga





## Lampiran 9 Tampilan halaman aspirasi

The screenshot shows a web browser window with the address bar displaying 'portalharga.net/aspirasi'. The website's header features the 'PortalHarga' logo and a navigation menu with links: Beranda, Cek Harga, Post Harga, Aspirasi, and Kadek. The main content area is titled 'Request Operasi Pasar' and contains a form with the following fields:

- Subjek\***: A text input field.
- Tanggal\***: A date input field with a placeholder 'mm/dd/yyyy'.
- Komoditas\***: A dropdown menu with the placeholder '-- Pilih Komoditas --'.
- Pasar\***: A dropdown menu with the placeholder '-- Pilih Lokasi Pasar --'.
- Pesan\***: A text area with a placeholder 'Ketikkan pesan Anda disini...'.

At the bottom right of the form are two buttons: 'Kirim' (orange) and 'Batal' (grey).

Lampiran 10 Kode html halaman *post* harga

```

23 <form class="col-md-4" id="form-add" method="post">
24   <input type="hidden" id="id_user"
    name="id_user" class="form-control" value="<?php echo $this-
    >session->userdata['user']['id_user'] ?>">
25
26   <div class="form-group" class="col-md-4">
27     <label for="komoditas">Komoditas<span
      class="required">*</span></label>
28     <select class="form-control" id="komoditas"
      name="id_komoditas">
29     <option value=""> -- Pilih Komoditas -- </option>
30     </select>
31   </div>
32
33   <div class="form-group" class="col-md-4">
34     <label for="harga_komoditas">Harga<span
      class="required">*</span></label>
35     <input class="form-control" type="text" id="harga"
      name="harga" placeholder="Masukkan harga tanpa simbol
      apapun, eg: 15000">
36   </div>
37
38   <div class="form-group" class="col-md-4">
39     <label for="tanggal">Tanggal<span
      class="required">*</span></label>
40     <input class="form-control" id="tanggal" name="tanggal"
      type="date">
41   </div>
42
43   <div class="form-group" class="col-md-4">
44     <label for="lokasi">Lokasi<span
      class="required">*</span></label>
45     <input class="form-control" name="lokasi" type="text"
      id="lokasi" readonly>
46   </div>
47
48   <div class="form-group" class="col-md-4">
49     <label for="longitude">Longitude<span
      class="required">*</span></label>
50     <input class="form-control" name="longitude" type="text"
      id="longitude" readonly>
51   </div>
52
53   <div class="form-group" class="col-md-4">
54     <label for="latitude">Latitude<span
      class="required">*</span></label>
55     <input class="form-control" name="latitude" type="text"
      id="latitude" readonly>
56   </div>
57
58   <!-- <div class="form-group" class="col-sm-14 text-
    right">
59     <button class="btn btn-action" type="button"
      onclick="add_post_harga()"> Post Harga </button>
60   </div> -->
61   <div class="modal-footer">

```

```
62         <button type="button" class="btn btn-primary"  
63             onclick="add_post_harga()">Post Harga</button>  
64         <button type="button" class="btn btn-default"  
65             onclick="cancel_post_harga()">Batal</button>  
66     </div>  
67 </form>  
68 <script src="http://maps.googleapis.com/maps/api/js?  
libraries=places"></script>
```

## **RIWAYAT HIDUP**

Penulis dilahirkan di Narmada pada tanggal 6 November 1994 dari pasangan bapak I Ketut Peggeg dan ibu Ni Wayan Sri Astuti. Penulis adalah putra kedua dari tiga bersaudara. Tahun 2012 penulis lulus dari SMA Negeri 1 Narmada dan pada tahun yang sama penulis lulus seleksi masuk Institut Pertanian Bogor melalui jalur SNMPTN Undangan dan diterima di Departemen Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam.

Pada tahun 2015, penulis melaksanakan Praktik Kerja Lapangan (PKL) di Kantor Komunikasi dan Informatika (Kominfo) Kota Bogor selama 35 hari kerja. Selama perkuliahan, penulis aktif dalam Unit Kegiatan Mahasiswa (UKM) Kesatuan Mahasiswa Hindu Dharma (KMHD) IPB, UKM Paduan Suara Mahasiswa IPB Agria Swara, dan Organisasi Mahasiswa Daerah Asal (OMDA) Bali.