

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

SHELLAFURI BIRU MARDIKA



**DEPARTEMEN ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
INSTITUT PERTANIAN BOGOR
BOGOR
2017**

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 Petani 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, Maret 2017

Shellafuri Biru Mardika
NIM G64120060

ABSTRAK

SHELLAFURI BIRU MARDIKA. Pengembangan Modul Pelaporan Harga Komoditas Pertanian pada Aplikasi Portal Harga untuk Petani Menggunakan REST API. Dibimbing oleh DEAN APRIANA RAMADHAN.

Harga produk komoditas pertanian dapat mengalami fluktuasi setiap harinya, salah satu penyebab terjadinya fluktuasi harga yaitu kehadiran tengkulak. Kehadiran tengkulak dapat menimbulkan kerugian pada pihak petani, hal ini dikarenakan adanya permainan harga oleh tengkulak. Penelitian ini merupakan pengembangan modul pelaporan harga komoditas pertanian untuk pengguna petani menggunakan *Representational State Transfer Application Programming Interface* (REST API) dengan struktur *database* MySQL. Metode yang digunakan dalam penelitian ini adalah *Extreme Programming*. Pengolahan data dilakukan pada REST API menggunakan Node.js dan hasil dari data tersebut disimpan dalam bentuk JSON. Penelitian ini bertujuan untuk memudahkan petani dalam menyampaikan hasil panen raya dan aspirasi kepada pemerintah. Selain itu, sistem ini juga memungkinkan petani untuk melihat aspirasi, melihat harga komoditas pertanian, melihat hasil panen raya dan mendukung aspirasi. Hasil pengujian dengan menggunakan metode *black box* menunjukkan bahwa fungsi utama aplikasi Portal Harga telah bekerja dengan baik sesuai dengan kebutuhan aplikasi.

Kata kunci: fluktuasi, harga, komoditas pertanian, REST API

ABSTRACT

SHELLAFURI BIRU MARDIKA. The Development of Price Reporting Module on Agriculture Commodities in Portal Harga Application for Farmers by Using REST API. Supervised by DEAN APRIANA RAMADHAN.

Agricultural commodities price may fluctuate every day, one of the causes is the middleman. The presence of middlemen may cause harm to the farmers, this is due to the price gaming by middlemen. This research is the development of agricultural commodity price reporting module for farmers side using the representational state transfer (REST) Application Programming Interface (API) with the MySQL database. The method used in this research is extreme programming. Data is processed on REST API using Node.js and the results of the data is saved in the form of JSON. This study aims to facilitate the farmers to present their harvest result and their aspirations to the government. In addition, the system also allows farmers to see aspirations, prices of agricultural commodities, and results of harvest and support aspirations. The testing result using black box method showed that the main function of Portal Harga application worked correctly according to the application requirements.

Keywords: agricultural commodities, fluctuation, price, REST API

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

SHELLAFURI BIRU MARDIKA

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

**DEPARTEMEN ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
INSTITUT PERTANIAN BOGOR
BOGOR
2017**

Penguji:

- 1 Husnul Khotimah, SKomp MKom
- 2 Rina Trisminingsih, SKomp MT

Judul Skripsi: Pengembangan Modul Pelaporan Harga Komoditas Pertanian pada Aplikasi
Portal Harga untuk Petani Menggunakan REST API
Nama : Shellafuri Biru Mardika
NIM : G64120060

Disetujui oleh



Dean Apriana Ramadhan, SKomp MKom
Pembimbing

Diketahui oleh



Dr Ir Agus Buono, MSi MKom
Ketua Departemen

Tanggal Lulus: **27 MAR 2017**

PRAKATA

Puji syukur kehadiran Allah subhanahu wa ta'ala yang telah melimpahkan rahmat dan hidayah-Nya, sehingga penulis mampu menyelesaikan penulisan skripsi yang berjudul Pengembangan Modul Pelaporan Harga Komoditas Pertanian pada Aplikasi Portal Harga untuk Petani Menggunakan REST API.

Skripsi ini tidak dapat tersusun dengan baik tanpa adanya bantuan dari berbagai pihak. Penulis mengucapkan terima kasih kepada Bapak Dean Apriana Ramadhan S.Komp, M.Kom selaku dosen pembimbing yang banyak memberikan arahan, nasehat, dan andil yang besar dalam penyelesaian skripsi penulis. Pada kesempatan ini penulis juga ingin mengucapkan terima kasih kepada:

- 1 Keluarga tercinta, Bapak Sumardi dan Ibu Holipah serta kakak tersayang, Siska Putri Jingga Mardika dan Handoyo yang selalu memberikan dorongan semangat, doa, dan nasehat kepada penulis selama menjalani perkuliahan hingga saat ini.
- 2 Ibu Husnul Khotimah, SKomp MKom dan Ibu Rina Trisminingsih, SKomp MT selaku dosen penguji yang telah memberikan kritik dan saran yang membangun dalam penelitian penulis.
- 3 Segenap jajaran dosen dan karyawan Departemen Ilmu Komputer atas ilmu dan bantuannya selama masa perkuliahan.
- 4 Team Portal Harga dan Kampanye, partner seperjuangan penelitian, Kadek, Didi, Dian, Indri, Reza Pahlevi dan Moh Reza Maskadi.
- 5 *My Best* (Defri, Pungky, Febi, Widuri, Wieke, Ramdhan, Alfandio dan Irfan) , teman-teman Wisma Nusantara (Lani, Fey, Ghvirly, Neneng, Rezky, Risa, Fina) dan segenap keluarga Ilmu Komputer angkatan 49 dan 50.
- 6 Semua pihak yang tidak dapat disebutkan satu persatu yang telah memberikan bantuan dalam kegiatan perkuliahan dan penyusunan skripsi ini.

Semoga karya ilmiah ini bermanfaat.

Bogor, Maret 2017

Shellafuri Biru Mardika

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	2
REST API	2
JSON	3
Express.JS	3
Roadmap Penelitian	3
METODE	5
Data Penelitian	5
Arsitektur Penelitian	5
Tahapan Penelitian	6
Perangkat Penelitian	7
HASIL DAN PEMBAHASAN	8
Data Penelitian	8
Tahapan Penelitian	8
SIMPULAN DAN SARAN	32
Simpulan	32
Saran	32
DAFTAR PUSTAKA	32
LAMPIRAN	34
RIWAYAT HIDUP	41

DAFTAR TABEL

1	Data aspirasi pesan	8
2	<i>User story</i>	9
3	<i>Fully developed use case description</i> fungsi <i>post</i> info panen raya	9
4	Pengujian fungsi REST iterasi pertama	17
5	Pengujian fungsi REST iterasi kedua	24
6	Pengujian REST iterasi ketiga	32

DAFTAR GAMBAR

1	Contoh format JSON	3
2	Roadmap penelitian <i>e-Government</i> Lab SEIS bidang pertanian	4
3	Arsitektur Penelitian	5
4	Proses dari model Extreme Programming (Pressman 2010)	6
5	Diagram <i>use case</i> iterasi pertama	8
6	<i>Swimlane</i> fungsi <i>post</i> info panen raya	10
7	<i>Class diagram</i> iterasi pertama	11
8	<i>Mockup</i> halaman pertama aplikasi portal harga	12
9	<i>Mockup</i> menu info panen	12
10	Potongan kode program untuk menghubungkan REST dengan basis data	13
11	Potongan kode REST fungsi <i>post</i> informasi hasil panen raya	14
12	<i>Output</i> REST <i>post</i> info panen raya	15
13	Potongan kode REST fungsi <i>post</i> aspirasi	16
14	<i>Output</i> REST fungsi <i>post</i> aspirasi	17
15	Diagram <i>use case</i> iterasi kedua	18
16	<i>Swimlane</i> fungsi melihat info harga komoditas pertanian	19
17	<i>Swimlane</i> fungsi melihat informasi hasil panen raya	19
18	<i>Class diagram</i> iterasi kedua	20
19	Implementasi halaman menu info panen	21
20	Implementasi halaman menu aspirasi	21
21	REST <i>post</i> informasi hasil panen iterasi kedua	22
22	Potongan kode untuk token	22
23	Contoh penggunaan token	23
24	Potongan AJAX <i>post</i> info panen	23
25	Potongan AJAX <i>get</i> komoditas	24
26	<i>Output</i> REST fungsi melihat info harga	25
27	<i>Output</i> REST fungsi melihat informasi hasil panen raya	26
28	<i>Use case diagram</i> iterasi ketiga	27
29	<i>Class diagram</i> iterasi ketiga	28
30	Implementasi halaman fungsi dukung aspirasi	28
31	Implementasi halaman fungsi batal dukung aspirasi	29
32	Potongan REST <i>post</i> dukung aspirasi	29
33	Potongan REST <i>get</i> dukung aspirasi	30
34	<i>Output</i> dari REST <i>get</i> dukung	30
35	Potongan AJAX untuk mendukung aspirasi	31

DAFTAR LAMPIRAN

1	<i>Fully developed use case description</i> aplikasi Portal Harga	34
2	<i>Swimlane</i> aplikasi Portal Harga	36
3	<i>Mockup</i> halaman aplikasi Portal Harga	37
4	Potongan kode REST	38

PENDAHULUAN

Latar Belakang

Sektor pertanian merupakan salah satu sektor yang penting di Indonesia. Menurut Sekjen Kemenperin (2013) pada Maret 2013 di Indonesia jumlah tenaga kerja yang bekerja di bidang pertanian mencapai 37.18 juta jiwa atau 32.61% dari jumlah tenaga kerja seluruhnya. Kebijakan-kebijakan terkait kesejahteraan petani harus diterapkan dengan tujuan agar dapat mendorong pertanian yang lebih produktif dan memberikan manfaat yang nyata bagi perekonomian di Indonesia.

Hal yang menyebabkan kurangnya tingkat kesejahteraan petani salah satunya adalah harga produk pertanian yang fluktuatif. Hal tersebut menyebabkan masyarakat merasa kesulitan untuk mencari pasar yang menjual hasil pertanian dengan harga yang lebih murah. Fluktuasi harga disebabkan karena jumlah pasokan dan permintaan yang dibutuhkan tidak seimbang (Hidayat *et al.* 2012). Penetapan harga oleh para tengkulak dinilai cukup merugikan para petani karena umumnya tengkulak yang menentukan harga yang kemudian diterima oleh petani dan petani tidak memiliki kesempatan untuk melakukan penawaran terhadap harga yang ditentukan oleh tengkulak tersebut. Hal tersebut menjadi sebuah permasalahan tersendiri yang hingga saat ini belum dapat terselesaikan.

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. Meskipun pemerintah sudah melakukan upaya perlindungan dan pemberdayaan terhadap petani, namun dalam praktiknya petani masih menghadapi berbagai masalah. Hal ini disebabkan karena kurangnya komunikasi dua arah antara pemerintah dan petani sehingga mengakibatkan keterlambatan penyampaian informasi dari petani ke pemerintah terkait hasil pertanian. Salah satu solusi adalah dengan mengembangkan aplikasi portal harga komoditas pertanian berbasis web untuk mendapatkan data yang cepat dan akurat.

Pembuatan aplikasi portal harga dilakukan secara tim yaitu dari penelitian sebelumnya telah dilakukan oleh Setiana (2016) yang mengerjakan modul admin dan pemerintah, Grihadevi (2016) mengerjakan modul masyarakat, dan Setyatama (2016) mengerjakan aplikasi portal harga pada platform *mobile android*. Modul admin memiliki fitur *create*, *read*, *update* dan *delete* (CRUD) manajemen *user*, CRUD manajemen komoditas, dan CRUD manajemen pasar. Modul pemerintah memiliki fitur memeriksa harga komoditas dan lokasi, melihat opini *request* operasi pasar, melihat aspirasi petani, rekapitulasi harga komoditas serta lokasi, dan melihat informasi hasil panen raya yang di masukkan oleh petani. Modul masyarakat memiliki fitur *post* harga komoditas pertanian, melihat harga komoditas pertanian, dan melakukan *request* operasi pasar. Penelitian ini dilakukan pada modul petani yang memfasilitasi petani untuk melaporkan informasi hasil panen raya kepada pemerintah, menyampaikan aspirasi secara langsung kepada pemerintah, mendukung aspirasi, melihat info panen raya dan info harga komoditas pertanian di pasar yang dimasukkan oleh masyarakat, mendukung aspirasi dan batal mendukung. Penelitian ini merupakan bagian dari *roadmap* penelitian *price and commodities agricultural e-Government* Lab SEIS.

Perumusan Masalah

Rumusan permasalahan pada penelitian ini yaitu bagaimana mengembangkan aplikasi web pelaporan harga komoditas pertanian berbasis REST API untuk petani yang dapat memfasilitasi petani dalam menyampaikan informasi hasil panen raya dan aspirasi kepada pemerintah.

Tujuan Penelitian

Tujuan dari penelitian ini yaitu mengembangkan aplikasi web pelaporan harga komoditas pertanian berbasis REST API untuk petani yang dapat memfasilitasi petani dalam menyampaikan informasi hasil panen raya dan aspirasi kepada pemerintah.

Manfaat Penelitian

Manfaat dari penelitian ini adalah:

- 1 Petani dapat menyampaikan informasi hasil panen dan aspirasi kepada pemerintah.
- 2 Petani dapat mengetahui informasi harga komoditas pertanian.

Ruang Lingkup Penelitian

Ruang lingkup penelitian ini adalah:

- 1 Komoditas pertanian yang digunakan terdiri atas daging sapi, bawang merah, dan cabai (cabai merah besar, cabai merah keriting, cabai rawit merah, dan cabai rawit hijau).
- 2 Aplikasi Portal Harga belum melakukan verifikasi data yang diberikan oleh pengguna.

TINJAUAN PUSTAKA

REST API

REpresentational State Transfer (REST) adalah salah satu jenis *web service* yang menerapkan konsep perpindahan antar *state* (Fauziah 2013). Penggunaan REST API 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). Dalam penelitian ini, sistem yang dibuat merupakan layanan berbasis web yang menggunakan arsitektur REST APIs. Perintah *HTTP* yang digunakan adalah fungsi *GET*, *POST*, *PUT*, dan *DELETE*. *Method GET* digunakan untuk mendapatkan data. *Method POST* digunakan untuk memasukkan data. *Method PUT*

digunakan untuk mengubah data. *Method DELETE* digunakan untuk menghapus data.

JSON

Javascript Object Notation (JSON) merupakan salah satu format pertukaran data ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer (JSON 2012). Penulisan JSON sangatlah singkat karena menggunakan notasi-notasi *javascript* seperti kurung siku [] dan kurung kurawal {}. Kurung siku pada [] JSON digunakan untuk membuat objek *array*, kurung kurawal {} digunakan untuk membuat sebuah objek baru. Kelebihan JSON adalah mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate* oleh komputer). Contoh penulisan format JSON dapat dilihat pada Gambar 1.

Gambar 1 merupakan contoh format JSON. *Array* JSON pada Gambar 1 yaitu *array* data komoditas yang terdiri atas tiga buah objek. Objek pertama terdiri atas atribut *id_komoditas* yang memiliki *value* 1, atribut *jenis* memiliki *value* beras, dan atribut *nama_komoditas* memiliki *value* beras. Objek kedua terdiri atas atribut *id_komoditas* yang memiliki *value* 2, atribut *jenis* memiliki *value* cabai, dan atribut *nama_komoditas* memiliki *value* cabai merah keriting. Objek ketiga terdiri atas atribut *id_komoditas* yang memiliki *value* 3, atribut *jenis* yang memiliki *value* cabai, dan atribut *nama_komoditas* yang memiliki *value* cabai rawit merah.

```
{
  "status": "success",
  "data": [
    {
      "id_komoditas": 1,
      "jenis": Beras,
      "nama_komoditas": "Beras"
    },
    {
      "id_komoditas": 2,
      "jenis": Cabai,
      "nama_komoditas": "Cabai merah keriting"
    },
    {
      "id_komoditas": 3,
      "jenis": Cabai,
      "nama_komoditas": "Cabai rawit merah"
    }
  ]
}
```

Gambar 1 Contoh format JSON

Express Js

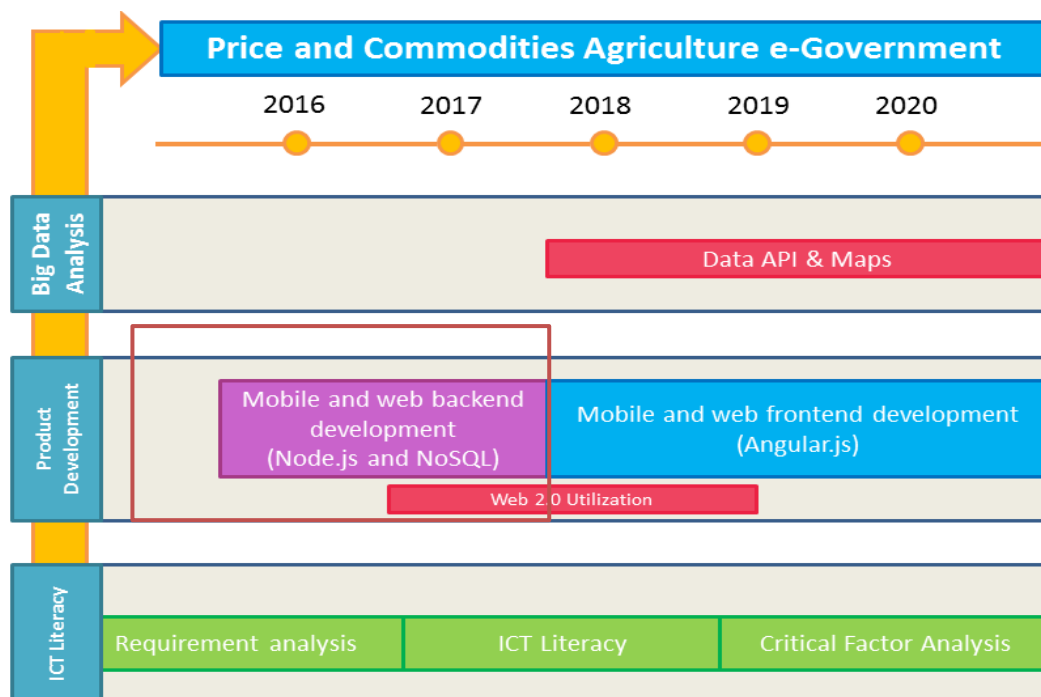
Express js merupakan salah satu *framework* Nodejs mempunyai komponen modul *http* dan *connect* (Mardan 2016). Komponen tersebut disebut dengan

middleware. Express js bekerja pada sebuah manajer paket dari node (NPM) sebagai modul dependensi untuk aplikasi. Oleh karena itu setiap proyek yang akan dibangun dengan Express.js perlu memiliki *file* sumber *framework* dalam folder *node_modules* lokal. Dengan *express.js programmer* dapat menambahkan *route*, memberikan otentifikasi, dan kode pemrograman yang lebih struktural.

Roadmap Penelitian

Roadmap merupakan rencana kerja rinci yang menggambarkan apa aja yang dilakukan untuk mencapai sebuah tujuan. *Roadmap* mengidentifikasi penelitian yang akan dilakukan berdasarkan jangka waktu penelitian dan prioritas serta mitra yang bisa diajak bekerjasama. Sehingga penelitian yang dilakukan akan terasa lebih bermakna, efektif dan efisien. Laboratorium *Software Engineering and Information Science* (SEIS) memiliki *roadmap* untuk mengembangkan *e-Government* dalam bidang pertanian untuk rentang waktu 2016-2020. Roadmap penelitian untuk *e-Government* ini memiliki tiga fokus kerja utama, yaitu *ICT literacy*, *product development*, dan *big data analysis*.

Penelitian ini merupakan pelaksanaan dari *product development* bagian *mobile and web backend development* pada *web 2.0 utilization*. Web 2.0 menyediakan interaksi antara pengguna dengan sistem. Pengguna dapat berpartisipasi dan berkolaborasi dalam mendistribusikan konten ke dalam web. Pada penelitian ini, pemanfaatan web 2.0 terdapat pada modul pelaporan informasi hasil panen raya dan penyampaian aspirasi yang ditandai dengan garis berwarna merah. Adapun *roadmap* penelitian dapat dilihat pada Gambar 2.



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

METODE

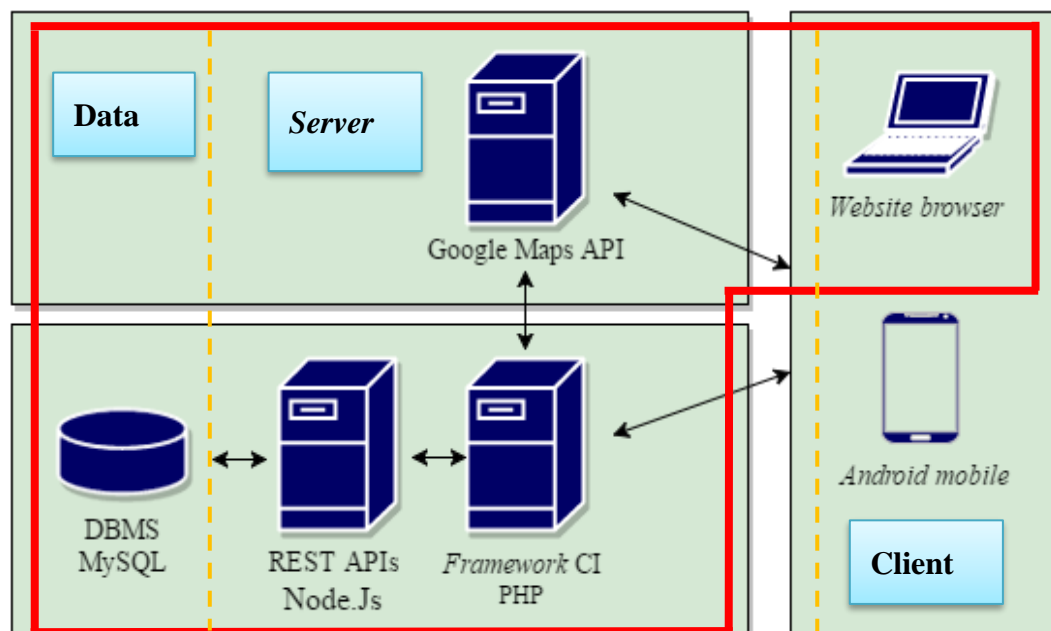
Data Penelitian

Data yang digunakan pada penelitian ini adalah data harga yang dimasukkan oleh masyarakat (Grihadevi 2016). Data panen raya dan aspirasi dimasukkan oleh pengguna yaitu petani. Pemilihan data komoditas pertanian berdasarkan dari komoditas yang mengalami fluktuasi setiap harinya yang tercantum pada situs <http://infopangan.jakarta.go.id>.

Arsitektur Penelitian

Arsitektur yang digunakan pada penelitian ini dapat dilihat pada Gambar 3 yang ditandai oleh kotak berwarna merah. Arsitektur pada penelitian ini terdiri atas tiga bagian yaitu:

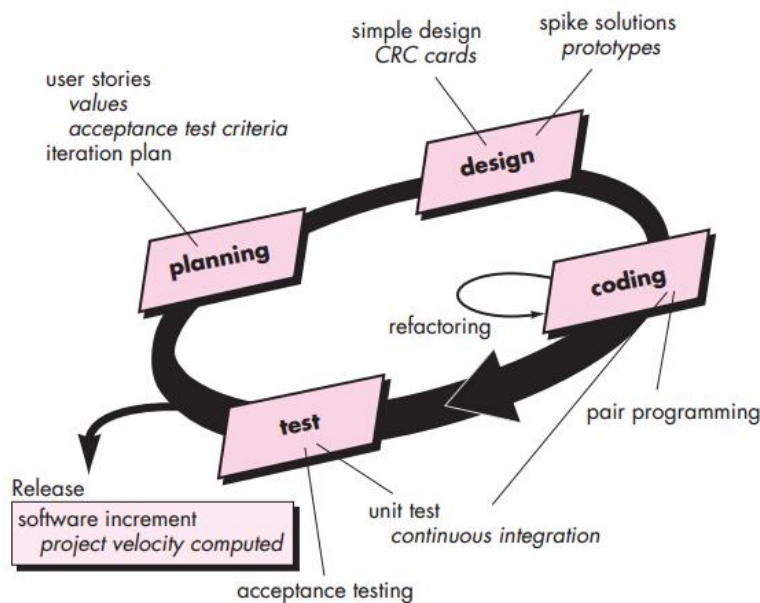
- 1 Bagian data, terdapat MySQL yang digunakan untuk basis data.
- 2 Bagian *client*, terdapat aplikasi berbasis web dan *mobile android*. Penelitian ini berfokus pada platform web dengan pengguna dari aplikasi web adalah petani.
- 3 Bagian *server*, terdapat *Google Maps API*, REST API dan *framework Codeigniter*. Fasilitas yang terdapat pada *Google Maps* antara lain adalah menjelajah peta, mencari lokasi tertentu dan mencari rute menuju suatu lokasi. Dalam penelitian ini *Google Maps API* digunakan untuk menampilkan peta dan menandai lokasi tempat petani yang mengalami panen raya. REST API digunakan sebagai perantara *request* dari *client* ke basis data. REST API menggunakan bahasa pemrograman Node.Js. *Framework Codeigniter* (CI) versi 3.03 dimanfaatkan untuk membuat tampilan halaman web.



Gambar 3 Arsitektur Penelitian

Tahapan Penelitian

Tahapan penelitian ini menggunakan metode *Extreme Programming* (XP) yang merupakan bagian dari *Agile Software Engineering*. Menurut Mohammadi *et al.* (2009), *Extreme Programming* merupakan metode yang paling terkenal karena dalam praktiknya lebih menekankan pada klient sehingga perubahan dapat dengan cepat ditanggapi. Adapun tahapan-tahapan yang dilakukan pada penelitian ini yaitu perencanaan, desain, pengkodean, dan pengujian. Gambar 4 menunjukkan tahapan pengembangan sistem menggunakan *XP Programming* menurut Pressman (2010). Penggunaan metode *XP Programming* sesuai dengan keadaan yang ada, karena penelitian ini dilakukan secara *tim* yang dibagi per modul.



Gambar 4 Proses dari Model Extreme Programming (Pressman 2010)

Perencanaan (XP Planning)

Tahap perencanaan dimulai dengan mengumpulkan *requirement*. Setelah semua terkumpul, dibuat *use case diagram*, *user story*, dan *fully developed use case description* dari aplikasi yang dibuat. *Use case diagram* dibuat untuk menggambarkan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat dan untuk mengetahui fungsi-fungsi utama yang terdapat dalam sistem. Pembuatan *user story* bertujuan untuk mempermudah *developer* dalam memahami keinginan dan kebutuhan pengguna, sehingga lebih mudah pula dalam melakukan tahapan selanjutnya. *User story* terdiri atas aktor yang menggunakan aplikasi, fungsi yang akan dibuat, deskripsi dari fungsi, dan prioritas kebutuhan pengguna. *Iteration plan* yaitu menentukan banyaknya iterasi yang dilakukan pada penelitian ini yaitu maksimal tiga kali iterasi.

Desain (Design)

Tahapan ini dilakukan pembuatan desain yang didasari dari perencanaan pada tahap sebelumnya. Dalam XP, proses desain dapat terjadi secara terus menerus selama proses pengembangan sistem dilakukan. Tahap desain dimulai dengan

pembuatan *class diagram* sebagai *class*, *responsibilities*, and *collaboration* (CRC). *Class diagram* digunakan untuk menggambarkan struktur dan deskripsi *class* serta hubungannya antar *class*. *Simple design* dalam XP dilakukan dengan pembuatan rancangan antarmuka (*mockup*) dari aplikasi yang dibuat. Implementasi dari *mockup* aplikasi Portal Harga sebagai pembuatan *spike solutions prototype*.

Pengkodean (XP Coding)

Setelah dilakukan tahapan perencanaan dan desain, maka tahapan selanjutnya adalah melakukan pengkodean (*coding*). Tahapan ini membuat modul secara satu per satu dan diuji setiap modul. Selain itu, di dalam Model XP terdapat *pair programming* yang memungkinkan lebih dari satu orang untuk melakukan pengkodean yang kemudian digabungkan dengan aplikasi utama. Pengkodean dilakukan dengan menggunakan bahasa pemrograman PHP, Node.js dan *framework Codeigniter* versi 3.03. *Refactoring* dalam XP berfokus pada penghapusan duplikasi dari kode yang telah dibuat. Penelitian ini melakukan *pair programming* dengan penelitian sebelumnya yang telah dilakukan oleh Grihadevi (2016) dan Setiana (2016).

Pengujian (Testing)

Pengujian dilakukan dengan menguji setiap fungsi pada modul. Setelah semua modul dikumpulkan dalam sebuah sistem yang sempurna, barulah pengujian penerimaan (*acceptance test*) dilakukan. Pada tahapan pengujian ini aplikasi akan diuji secara internal oleh tim dengan menggunakan metode pengujian *black-box*. Metode *black-box* merupakan metode pengujian yang hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak (Pressman 2010).

Perangkat Penelitian

Perangkat keras berupa komputer personal dengan spesifikasi sebagai berikut:

- 1 Processor Intel Core i3-4030U
- 2 Memori RAM 2 GB
- 3 *Keyboard*
- 4 *Mouse*

Perangkat lunak :

- 1 Sistem operasi Windows 10
- 2 Bahasa pemrograman HTML, Javascript, NodeJs, dan PHP
- 3 *Framework CodeIgniter* versi 3.03
- 4 Sublime text dan Atom sebagai *text editor*
- 5 Apache2 sebagai *web server*
- 6 DBMS MySQL
- 7 Postman digunakan untuk menguji REST API

HASIL DAN PEMBAHASAN

Data Penelitian

Data yang digunakan adalah data yang bersumber dari pengguna aplikasi yang menyampaikan informasi tentang hasil panen raya dan aspirasi yang dikirim petani ke pemerintah. Selain itu, juga terdapat data harga yang dimasukkan oleh masyarakat (Grihadevi 2016). Contoh data aspirasi dapat dilihat pada Tabel 1. Pada data aspirasi terdapat *field* aspirasi_pesanan dan *time*. *Field* aspirasi_pesanan merupakan informasi aspirasi yang dimasukkan oleh pengguna. *Field* *time* merupakan *field* yang berisi informasi waktu dikirimnya data aspirasi ke pemerintah.

Tabel 1 Data aspirasi pesan

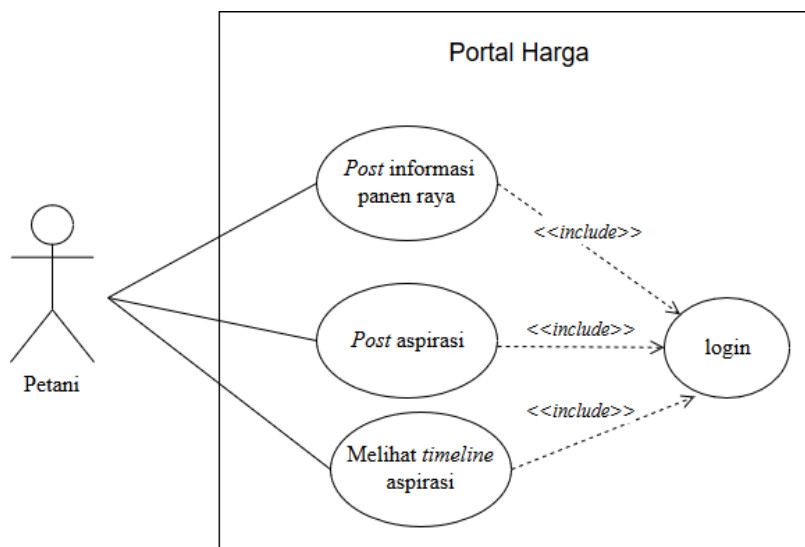
No	Aspirasi_pesanan	Time
1	Perlu diadakan sosialisasi di daerah dramaga	2015-05-29 22:58:03.9

Tahapan Penelitian

Iterasi Pertama

Perencanaan Iterasi Pertama (XP Planning)

Tahap perencanaan diawali dengan pembuatan diagram *use case* yang dapat dilihat pada Gambar 5. Diagram *use case* ini menggambarkan interaksi antara aktor dengan sistem yang dibangun. Pada diagram *use case*, aplikasi portal harga mempunyai tiga fungsi, yaitu memasukkan informasi hasil panen raya, memasukkan aspirasi dan melihat *timeline* aspirasi.



Gambar 5 Diagram *use case* iterasi pertama

Selain diagram *use case*, tahapan perencanaan juga menghasilkan *user story* dan *fully developed use case description*. Tabel 2 merupakan *user story* dari aplikasi

Portal Harga. Pada *user story* terdapat prioritas yang diurutkan berdasarkan tingkat kepentingan *task* yang harus dibuat.

Tabel 2 *User story*

No	Task	User story	Prioritas
1	<i>Post info panen raya</i>	Petani melaporkan informasi panen raya kepada pemerintah dengan memasukkan nama komoditas, tanggal panen raya, foto komoditas dan lokasi.	1
2	<i>Post aspirasi</i>	Petani mempunyai aspirasi terkait pertanian dan menyampaikan kepada pemerintah.	2
3	Melihat <i>timeline</i> aspirasi	Petani dapat melihat <i>timeline</i> aspirasi.	3

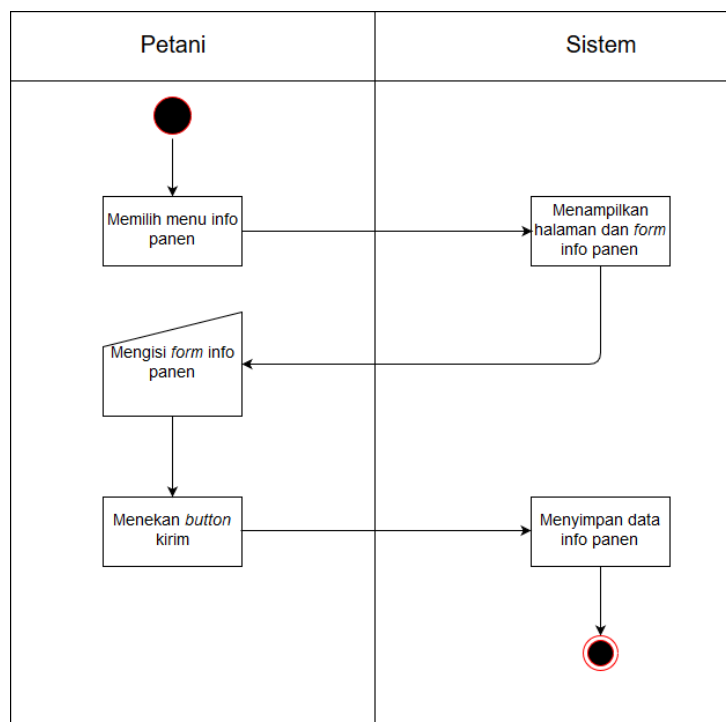
Fully developed use case description memuat penjelasan lebih rinci tentang skenario, deskripsi, aktor, aktivitas dari *use case*. *Fully developed use case description* fungsi *post info panen raya* dapat dilihat pada Tabel 3. *Fully developed use case description* untuk fungsi *post aspirasi* dan fungsi melihat *timeline* aspirasi dapat dilihat pada Lampiran 1.

Tabel 3 *Fully developed use case description* fungsi *post info panen raya*

<i>Use case</i>	Deskripsi	
<i>Name</i>	<i>Post info panen raya</i>	
<i>Scenario</i>	Petani ingin menyampaikan hasil panen raya kepada pemerintah	
<i>Triggering Event</i>	Petani berencana untuk melakukan panen raya dan akan melaporkannya kepada pemerintah.	
<i>Brief Description</i>	Petani menyampaikan hasil panen raya kepada pemerintah. Petani menggunakan aplikasi portal harga untuk mengirimkan data hasil panen raya kepada pemerintah.	
<i>Actors</i>	Petani	
<i>Related Use Case</i>	Login	
<i>Stakeholders</i>	Petani	
<i>Preconditions</i>	Petani telah berhasil melakukan <i>login</i> dan memilih menu info panen	
<i>Post conditions</i>	Petani berhasil memasukkan data hasil panen.	
<i>Flow of Events</i>	<i>Actor</i>	<i>System</i>
	1 Memilih menu info panen	1.1 Menampilkan halaman menu info panen
	2 Mengisi form info panen yang terdiri atas komoditas yang dipanen, tanggal perkiraan awal dan akhir panen, jumlah perkiraan komoditas yang dipanen, keterangan, foto komoditas, lokasi beserta longitude dan latitude.	2.1 Menyimpan data info panen yang dimasukkan pengguna

Use case	Deskripsi
Exceptions	- <i>Form</i> info panen tidak diisi secara lengkap, petani tidak bisa melakukan <i>post</i> info panen.
Conditions	- Jika ingin melakukan <i>post</i> info panen, petani harus melakukan <i>login</i> terlebih dahulu

Pembuatan *swimlane* pada tahap ini bertujuan untuk menggambarkan alur interaksi antara pengguna dengan sistem. Pada tahap ini dilakukan pembuatan *swimlane* untuk fungsi *post* info panen raya, *post* aspirasi, dan melihat *timeline* aspirasi. *Swimlane* untuk *post* info panen raya dapat dilihat Gambar 6 yang menunjukkan ketika pengguna ingin memasukkan info panen raya, pengguna harus memilih menu info panen, dan sistem akan menampilkan halaman menu info panen dan menyediakan *form* info panen yang dapat diisi oleh pengguna. Ketika pengguna berhasil memasukkan info panen, sistem akan menyimpan dan mengirimkan data ke *database* melalui REST. *Swimlane* untuk *post* aspirasi dan melihat *timeline* aspirasi dapat dilihat pada Lampiran 2.



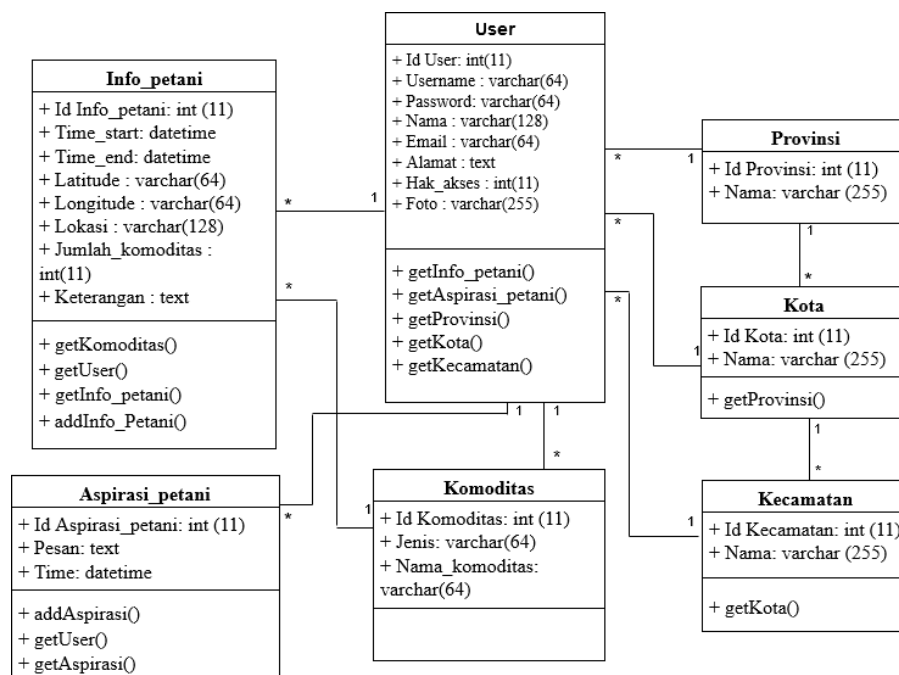
Gambar 6 *Swimlane* fungsi *post* info panen raya

Desain Iterasi Pertama (XP Design)

Tahap desain dimulai dengan pembuatan *mockup* dan perancangan *database* dari aplikasi yang dibangun. Perancangan *database* dimulai dengan pembuatan *class diagram* sebagai CRC dalam *XP Programming* yang bertujuan untuk menggambarkan struktur dan deskripsi *class* serta hubungan antar *class*. Pada penelitian ini terdapat 7 kelas meliputi *class user*, *info_petani*, *aspirasi_petani*, *komoditas*, *provinsi*, *kota* dan *kecamatan*. Masing-masing *class* tersebut memiliki *method* dan atribut berbeda. *Class diagram* untuk sistem yang dibangun dapat dilihat pada Gambar 7.

Pada Gambar 7 *class user* digunakan untuk menyimpan data pengguna yang memiliki atribut *id_user*, *username*, *password*, nama, email, alamat, foto dan hak akses. *Class info_petani* digunakan untuk menyimpan data info panen raya terdiri atas atribut *id_info_petani*, atribut *time_start* digunakan untuk menyimpan waktu perkiraan awal panen raya, atribut *time_end* digunakan untuk menyimpan waktu perkiraan akhir panen raya, atribut jumlah komoditas, atribut keterangan dan atribut latitude dan longitude digunakan untuk mengambil lokasi dari pengguna. *Class komoditas* digunakan untuk menyimpan data komoditas yang terdiri atas atribut *id_komoditas*, jenis dan nama_komoditas. *Class aspirasi_petani* digunakan untuk menyimpan data aspirasi yang dimasukkan oleh pengguna yang terdiri atas atribut *id_aspirasi_petani*, pesan, dan *time*. *Class provinsi*, kelas kota, dan kelas kecamatan digunakan untuk mengambil alamat dari pengguna yang masing-masing terdiri atas atribut *id* dan nama.

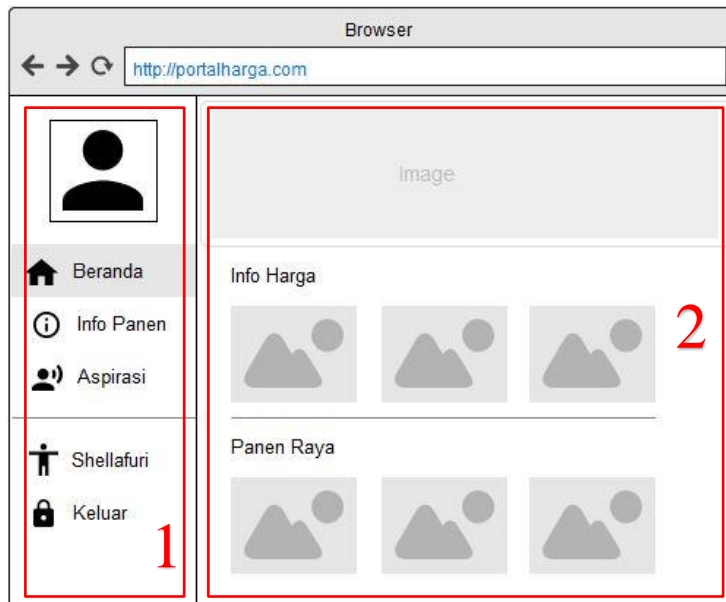
Class diagram pada Gambar 7 juga menunjukkan hubungan antar *class*. *Class user* memiliki hubungan dengan *class info_petani* dengan nilai kardinalitas satu ke banyak, hal ini menunjukkan bahwa satu pengguna dapat memasukkan informasi hasil panen lebih dari satu kali. *Class user* memiliki hubungan dengan *class provinsi*, *class kota* dan *class kecamatan* yang masing-masing memiliki nilai kardinalitas banyak ke satu, nilai kardinalitas ini menunjukkan bahwa satu provinsi satu kota dan satu kecamatan ditempati oleh banyak user. *Class user* dengan *class aspirasi_petani* memiliki nilai kardinalitas satu ke banyak. Hal ini menunjukkan bahwa satu pengguna bisa mengirimkan aspirasi lebih dari satu.



Gambar 7 *Class diagram* iterasi pertama

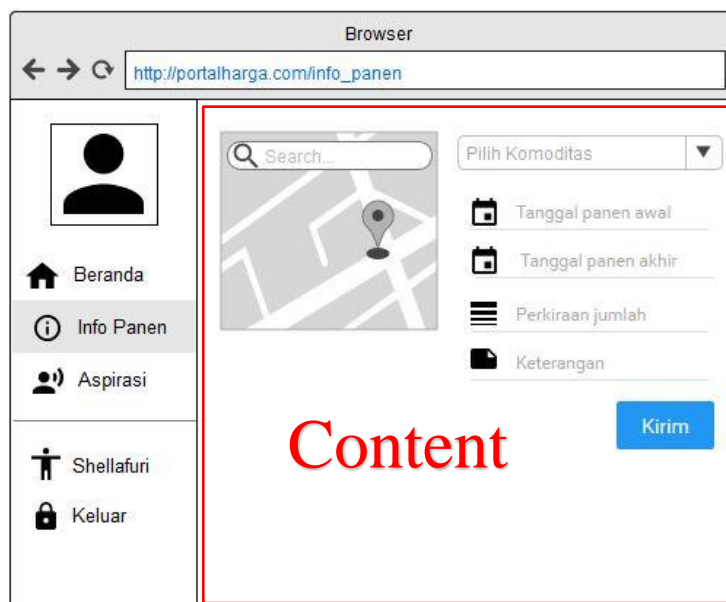
Simple design pada *XP Programming* dilakukan dengan membuat perancangan antarmuka (*mockup*) untuk semua fungsi. Gambar 8 menunjukkan *mockup* untuk halaman pertama aplikasi portal harga. *Mockup* dari halaman aplikasi portal harga terdiri atas dua bagian yang dibatasi oleh garis merah dengan masing-masing ditandai oleh angka 1 dan 2 pada Gambar 8. Bagian (1) merupakan bagian

side bar yang terdiri atas menu navigasi yaitu menu beranda, info panen, aspirasi, profil, dan keluar, sementara itu pada bagian (2) merupakan bagian *content*. Bagian *content* pada halaman pertama aplikasi portal harga terdiri atas fitur info harga dan informasi hasil panen raya.



Gambar 8 *Mockup* halaman pertama aplikasi portal harga

Gambar 9 menunjukkan *mockup* untuk menu info panen. Pada Gambar 9 bagian *content* terdiri atas *form* info panen dan peta untuk menandai lokasi dari pengguna. *Mockup* halaman *post* aspirasi dan halaman melihat *timeline* aspirasi dapat dilihat pada Lampiran 3.



Gambar 9 *Mockup* menu info panen

Pengkodean Iterasi Pertama (XP Coding)

Tahap ini dilakukan pengkodean untuk menghubungkan REST dengan basis data yang dapat dilihat pada Gambar 10. Potongan kode pada Gambar 10 menunjukkan REST yang dihubungkan ke basis data MySQL dengan *localhost* sebagai *host*, *user* menggunakan *root*, *password* dikosongkan dan *database* yang digunakan bernama *portal*. *Api_portalharga* menunjukkan *base route* yang digunakan. Ketika *server* berhasil dijalankan, *server* akan menampilkan pesan “All right ! I am alive at Port 3000.”

```
function REST(){
    var self = this;
    self.connectMysql();
};
REST.prototype.connectMysql = function(){
    var self = this;
    var pool = mysql.createPool({
        host      : 'localhost',
        user      : 'root',
        password  : '',
        database  : 'portal'
    });
    pool.getConnection(function(err,connection){
        if(err) {
            self.stop(err);
        } else {
            self.configureExpress(connection);
        }
    });
}
REST.prototype.configureExpress = function(connection){
    var self = this;
    app.use(bodyParser.urlencoded({ extended: true }));
    app.use(bodyParser.json());
    var router = express.Router();
    app.use('/api_portalharga', router);
    var rest_router = new rest(router,connection,md5);
    self.startServer();
}
REST.prototype.startServer = function(){
    app.listen(3000,function(){
        console.log("All right ! I am alive at Port 3000.");
    });
}
REST.prototype.stop = function(err){
    console.log("ISSUE WITH MYSQL \n" + err);
    process.exit(1);
}
```

Gambar 10 Potongan kode program untuk mengubungkan REST dengan basis data

Setelah menghubungkan REST dengan basis data, selanjutnya dilakukan pembuatan REST API dengan method *POST*, *DELETE*, *PUT*, dan *GET* dengan menggunakan bahasa pemrograman node.js. Potongan kode REST API untuk

fungsi *post* info panen dapat dilihat pada Gambar 11. *Method POST* pada Gambar 11 digunakan untuk memasukkan data ke dalam tabel *info_petani*. Bagian yang ditandai oleh garis merah pada Gambar 11 digunakan untuk melakukan proses *query* ke *database*. Bagian A pada Gambar 11 digunakan untuk atribut-atribut yang ada pada tabel *info_petani*. Bagian B pada Gambar 11 menunjukkan nilai dari atribut yang didapat dari *request body* pada *form* info panen raya. Jika *query* berhasil, *server* akan mengirimkan pesan “Berhasil mengirim info panen”, sedangkan jika terjadi eror karena kesalahan *query* akan muncul pesan “Error executing MySQL query”.

```
router.post("/info_petani", function(req,res){
  var query="INSERT INTO ??(??,??,??,??,??,??,??,??)VALUES
  (?,?,?,?,?,?,?,?)";
  var table = ["info_petani", "id_komoditas", "id_user",
    "time", "latitude", "longitude", "lokasi",
    "jumlah_komoditas", "keterangan",
    req.body.id_komoditas, req.body.id_user, new
    Date(req.body.time), req.body.latitude,
    req.body.longitude, req.body.lokasi,
    req.body.jumlah_komoditas, req.body.keterangan];
  query = mysql.format(query,table);
  connection.query(query,function(err,rows){
    if(err)
    {
      res.json({"status" : "error" , "message" : "Error
      executing MySQL query"});
    }
    else
    {
      res.json({"status" : "success" , "message" : "Berhasil
      mengirim info panen"
    });
    }
  });
});
```

Gambar 11 Potongan kode REST fungsi *post* informasi hasil panen raya

Gambar 12 merupakan *output* dari fungsi *post* info panen raya. *Output* berupa data dalam format JSON. Array JSON pada Gambar 12 yaitu *array* data hasil panen yang memiliki satu objek yang terdiri atas 19 atribut. Atribut *id_info_petani* memiliki *value* 45, hal ini menunjukkan bahwa data info petani yang dimasukkan memiliki id 45. Atribut *id_komoditas* memiliki *value* 2, hal ini menunjukkan bahwa komoditas yang dipanen memiliki id 2 yaitu komoditas cabai merah keriting. Atribut *id_user* memiliki *value* 9, hal ini menunjukkan bahwa pengguna yang memasukkan hasil panen raya memiliki identitas unik 9. Atribut *time_start* memiliki *value* 2017-01-26 00:00:00.000, hal ini menunjukkan bahwa perkiraan waktu awal pengguna mengalami panen raya yaitu tanggal 26 Januari 2017. Atribut *time_end* memiliki *value* 2017-01-30 00:00:00.000, hal ini menunjukkan bahwa perkiraan waktu akhir pengguna mengalami panen raya yaitu tanggal 30 Januari 2017. Atribut lokasi memiliki *value* Depok City, West Java, Indonesia, hal ini

menunjukkan bahwa lokasi pengguna yang memasukkan info hasil panen raya berada di Kota Depok. Atribut *latitude* dan *longitude* memiliki *value* masing-masing -6.4024844 dan 106.7942405, *value* dari atribut *latitude* dan *longitude* ini menunjukkan koordinat geografis dari Kota Depok. Atribut *jumlah_komoditas* memiliki *value* 26, hal ini menunjukkan bahwa perkiraan jumlah komoditas yang dipanen sebanyak 26 ton. Atribut *keterangan* memiliki *value* cabai Merah Keriting masih segar, atribut *keterangan* ini merupakan atribut opsional. Atribut *username* memiliki *value* shellafuribiru, hal ini menunjukkan bahwa pengguna yang memasukkan info hasil panen raya memiliki *username* yaitu shellafuribiru. Atribut *password* menunjukkan *password* dari akun pengguna. Atribut *nama* dan *email* digunakan untuk identitas dari pengguna yang masing-masing memiliki *value* shellafuri biru mardika dan shella@gmail.com. Atribut *id_provinsi*, *id_kota*, dan *id_kecamatan* menunjukkan lokasi alamat dari pengguna yang memasukkan info panen raya, masing-masing atribut ini memiliki *value* 32, 3276 dan 3276021. *Value* 32 menunjukkan provinsi Jawa Barat, *value* 3276 kota Depok, dan *value* 3276021 menunjukkan kecamatan Cipayung. Atribut *alamat* memiliki *value* perumahan graha mas, Cipayung, Depok. Atribut *hak_akses* memiliki *value* 4, atribut ini menunjukkan kewenangan yang diberikan bagi pengguna ketika *login*.

```
{
  "status": "success",
  "message": "Berhasil mengirim info panen",
  "data": [
    {
      "id_info_petani": 45,
      "id_komoditas": 2,
      "id_user": 9,
      "time_start": "2017-01-26 00:00:00.000",
      "time_end": "2017-01-30 00:00:00.000",
      "lokasi": "Depok City, West Java, Indonesia",
      "latitude": "-6.4024844",
      "longitude": "106.7942405",
      "jumlah_komoditas": "26",
      "keterangan": "Cabai Merah Keriting masih segar",
      "username": "Shellafurii",
      "password": "cffbad68bb97a6c3f943538f119c992c",
      "nama": "shellafuri biru mardika",
      "email": "shella@gmail.com",
      "id_provinsi": 32,
      "id_kota": 3276,
      "id_kecamatan": 3276021,
      "alamat": "Perumahan graha mas, Cipayung, Depok",
      "hak_akses": 4
    }
  ]
}
```

Gambar 12 Output REST *post* info panen raya

Potongan kode REST untuk aspirasi dapat dilihat pada Gambar 13. Bagian yang ditandai oleh garis merah pada Gambar 13 digunakan untuk melakukan *query*

ke *database*. Bagian A pada Gambar 13 digunakan untuk atribut-atribut yang ada pada tabel aspirasi. Bagian B pada Gambar 13 menunjukkan nilai dari atribut yang di dapat dari *request body* pada *form* aspirasi. Jika proses *query* gagal maka *server* akan memberikan pesan “Error executing MySQL query”. Proses *query* berhasil maka *server* akan memberikan pesan “Berhasil membuat aspirasi” dan akan menampilkan data dengan format JSON.



```
router.post("/aspirasi_petani", function(req, res) {
  var time = new Date();

  var query = "INSERT INTO ??(A,?, ?, ?) VALUES (B,?, ?, ?)";
  var table = ["aspirasi_petani", "id_user", "aspirasi_pesan",
    "time", req.body.id_user, req.body.aspirasi_pesan, time];
  query = mysql.format(query, table);

  connection.query(query, function(err, rows) {

    if (err) {
      res.json({
        "status": "error",
        "message": "Error executing MySQL query"
      });
    } else {
      res.json({
        "status": "success",
        "message": "Berhasil membuat aspirasi",
        "data": rows
      });
    }
  });
});
```

Gambar 13 Potongan kode REST fungsi *post* aspirasi

Gambar 14 merupakan *Output* REST dari fungsi *post* aspirasi. *Array* JSON pada Gambar 14 adalah data aspirasi yang memiliki satu objek yang terdiri atas 14 atribut. Atribut *id_aspirasi* memiliki *value* 58, hal ini menunjukkan bahwa aspirasi yang dimasukkan memiliki id 58. Atribut *id_user* memiliki *value* 7, hal ini menunjukkan bahwa pengguna yang memasukkan aspirasi memiliki id 7. Atribut *aspirasi_pesan* digunakan untuk pesan aspirasi yang dimasukkan oleh pengguna. Atribut *time* memiliki *value* 2017-01-26 18:58:05.883, hal ini menunjukkan bahwa waktu aspirasi yang dimasukkan yaitu pada tanggal 26 Januari 2017. Atribut *username*, *password*, *nama*, dan *email* merupakan atribut yang menunjukkan identitas dari pengguna yang memasukkan aspirasi. Atribut *username* memiliki *value* kadekgrihadevi, atribut *password* memiliki *value* hasil enkripsi *password* yaitu 21232f297a57a5a743894a0e4a801fc3, atribut *nama* memiliki *value* kadek, atribut *email* memiliki *value* kadek@gmail.com. Atribut *id_provinsi*, *id_kota*,

id_kecamatan dan alamat merupakan atribut yang menunjukkan lokasi alamat dari pengguna yang memasukan aspirasi. Atribut id_kota memiliki *value* 32, *value* ini menunjukkan provinsi jawa barat. Atribut id_kota memiliki *value* 3201, *value* ini menunjukkan kabupaten bogor. Atribut id_kecamatan memiliki *value* 32, *value* ini menunjukkan provinsi jawa barat. Atribut id_kecamatan memiliki *value* 3201300, *value* ini menunjukkan kecamatan parung panjang. Atribut alamat memiliki *value* Desa Parungpanjang, Kec. Parungpanjang, Kab. Bogor, Jawa Barat. Atribut foto memiliki *value* default.jpg, *value* ini menunjukkan *path* foto yang digunakan. Atribut hak_akses memiliki *value* 4, atribut ini menunjukkan kewenangan yang diberikan bagi pengguna ketika *login*.

```
{
  "status": "success",
  "message": "Berhasil membuat aspirasi",
  "data": [
    {
      "id_aspirasi": 58,
      "id_user": 7,
      "aspirasi_pesan": "mohon diadakan penyuluhan di daerah dramaga",
      "time": "2017-01-26 18:58:05.883",
      "username": "kadekgrihadevi",
      "password": "21232f297a57a5a743894a0e4a801fc3",
      "nama": "kadek",
      "email": "kadek@gamil.com",
      "id_provinsi": 32,
      "id_kota": 3201,
      "id_kecamatan": 3201300,
      "alamat": "Desa Parungpanjang, Kec. Parungpanjang, Kab. Bogor, Jawa Barat",
      "foto": "default.jpg",
      "hak_akses": 4
    }
  ]
}
```

Gambar 14 *Output* REST fungsi *post* aspirasi

Pengujian Iterasi Pertama (XP Testing)

Tahapan pengujian dilakukan dengan menggunakan *software* Postman yang dilakukan oleh tim internal. Tabel 4 merupakan pengujian REST API untuk fungsi *post* info panen raya, fungsi *post* aspirasi dan fungsi melihat *timeline* aspirasi

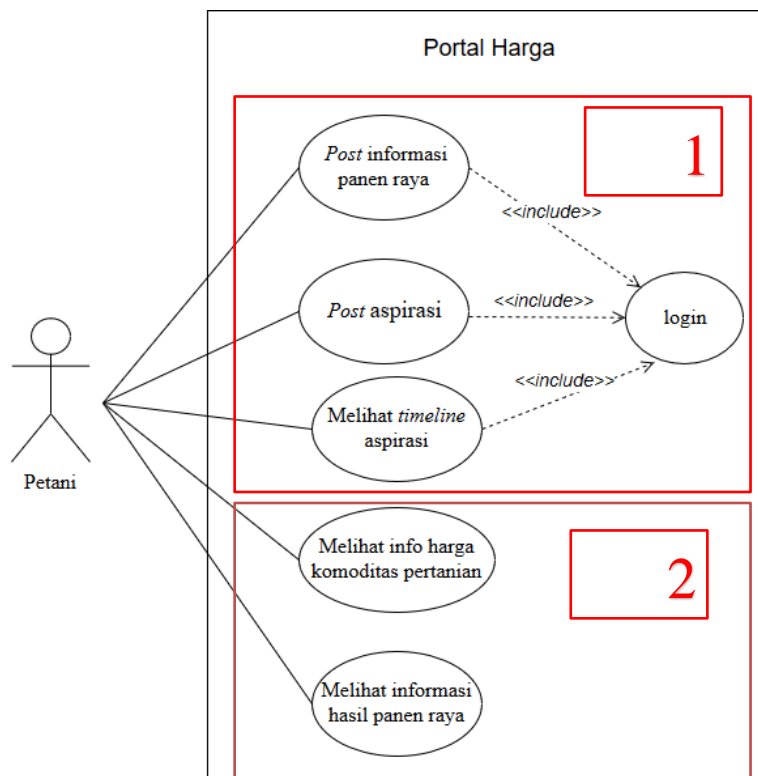
Tabel 4 Pengujian fungsi *REST* iterasi pertama

Aktor	Fungsi REST	Status Pengujian
Petani	<i>Post</i> info panen raya	Berhasil
	<i>Post</i> aspirasi	Berhasil
	Melihat <i>timeline</i> aspirasi	Berhasil

Iterasi Kedua

Perencanaan Iterasi Kedua (XP Planning)

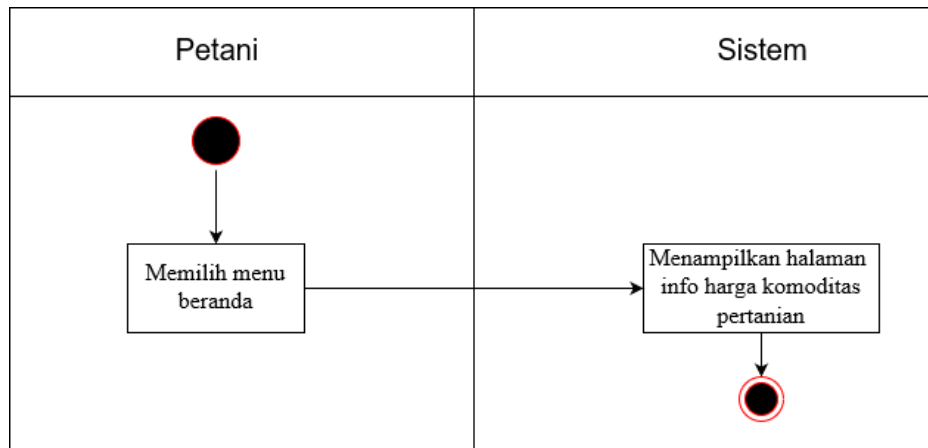
Tahap perencanaan pada iterasi kedua dilakukan perbaikan *use case* dan *fully developed use case story*. Gambar 15 menunjukkan perbaikan diagram *use case*. Bagian (1) pada Gambar 15 merupakan fungsi yang sudah dikerjakan pada iterasi pertama. Perbaikan diagram *use case* pada tahap perencanaan iterasi kedua terdapat pada Gambar 15 bagian (2) yaitu dengan penambahan fungsi melihat info harga komoditas pertanian dan fungsi melihat informasi panen raya. Petani tidak perlu melakukan login untuk melakukan fungsi melihat info harga komoditas pertanian dan fungsi melihat informasi hasil panen raya.



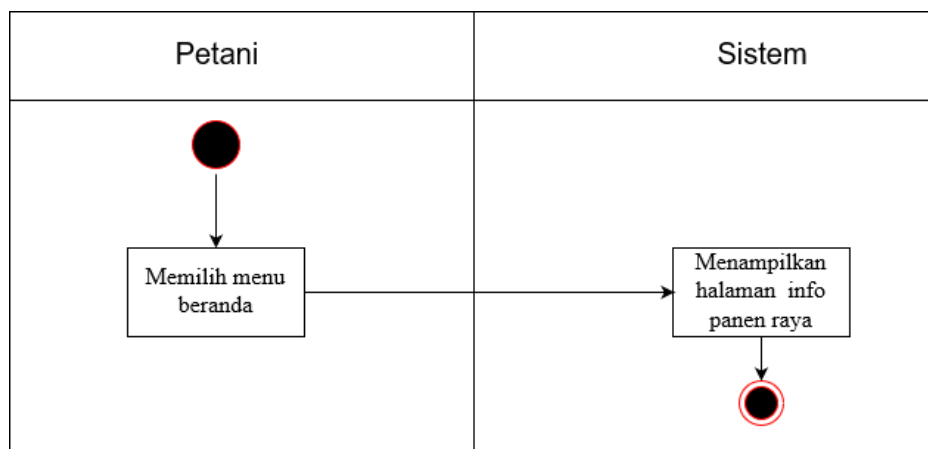
Gambar 15 Diagram *use case* iterasi kedua

Tahap perencanaan iterasi kedua juga dilakukan pembuatan *swimlane* untuk fungsi melihat info harga komoditas pertanian dan melihat informasi hasil panen raya. Gambar 16 merupakan *swimlane* dari fungsi melihat info harga komoditas pertanian. Bagian yang terlibat dalam fungsi melihat info harga komoditas yaitu aktor petani dan sistem. Petani memilih menu beranda dan sistem akan menampilkan halaman info harga komoditas pertanian.

Swimlane untuk fungsi melihat informasi hasil panen raya dapat dilihat pada Gambar 17. Petani memilih menu beranda dan sistem akan menampilkan halaman info panen raya. *Fully developed use case story* untuk fungsi melihat info harga komoditas pertanian dan melihat informasi hasil panen raya dapat dilihat pada Lampiran 1.



Gambar 16 *Swimlane* fungsi melihat info harga komoditas pertanian

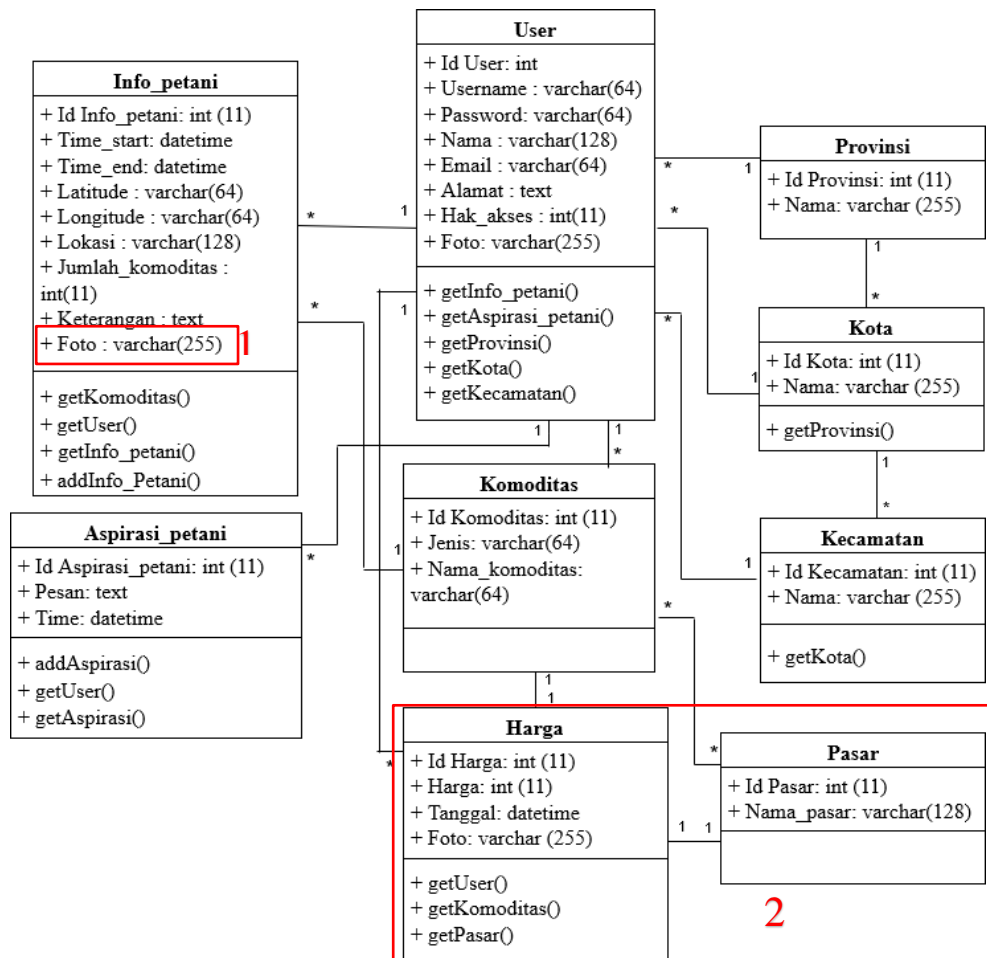


Gambar 17 *Swimlane* fungsi melihat informasi hasil panen raya

Desain Iterasi Kedua (XP Design)

Tahap desain pada iterasi kedua dimulai dengan melakukan perbaikan *class diagram*. Perbaikan *class diagram* iterasi kedua dapat dilihat pada Gambar 18 yang ditandai dengan garis merah dan diberi angka 1 dan 2. Bagian (1) menunjukkan perbaikan *class diagram* dengan penambahan atribut foto dengan tipe data *varchar* (255) pada tabel *info_petani*. Atribut foto ini digunakan untuk mengunggah foto komoditas yang dipanen.

Perbaikan *class diagram* juga dilakukan dengan penambahan *class* harga dan *class* pasar yang ditandai dengan garis merah pada bagian (2) dalam Gambar 18. *Class* harga digunakan untuk menyimpan data harga komoditas. *Class* pasar digunakan untuk menyimpan data pasar. *Class* komoditas dengan kelas harga memiliki hubungan dengan nilai kardinalitas satu ke satu, hal ini menunjukkan bahwa satu komoditas hanya memiliki satu harga. *Class* user dengan kelas harga memiliki hubungan dengan nilai kardinalitas satu ke banyak, hal ini menunjukkan bahwa satu *user* dapat memasukkan lebih dari satu info harga suatu komoditas pertanian. *Class* harga dengan kelas pasar memiliki hubungan dengan nilai kardinalitas satu ke satu, hal ini menunjukkan satu harga komoditas terdapat pada satu pasar.



Gambar 18 Class diagram iterasi kedua

Pembuatan tampilan halaman terdiri atas halaman beranda, halaman info panen halaman aspirasi, dan halaman profil. Untuk keseluruhan halaman pada penelitian ini terdiri dua bagian yaitu bagian *side bar* dan *content*. *Side bar* berisi menu navigasi yang terdiri atas menu beranda, info panen, aspirasi, profil, dan *logout*. Bagian *content* berisi *content* dari masing-masing menu. Menu beranda berisi informasi harga komoditas pertanian dan informasi hasil panen raya. Menu info panen merupakan menu untuk pengguna memasukkan informasi hasil panen raya. Menu aspirasi merupakan menu untuk pengguna mengirimkan aspirasi kepada pemerintah dan menampilkan *timeline* aspirasi. Menu profil berisi identitas dari pengguna. *Logout* digunakan ketika petani ingin keluar dari aplikasi.

Gambar 19 menunjukkan implementasi halaman menu info panen. Bagian *content* untuk menu info panen terdiri atas peta dan *form*. Peta digunakan untuk menandai lokasi pengguna mengalami panen. *Field* untuk *form* info panen terdiri *field* komoditas, tanggal perkiraan awal panen, tanggal perkiraan akhir panen, jumlah komoditas, keterangan, dan foto komoditas.

Gambar 20 merupakan implementasi halaman menu aspirasi. Bagian *content* menu aspirasi terdiri atas *form* pesan aspirasi yang terdiri dari *field* aspirasi_pesan dan *timeline* aspirasi. Ketika pengguna mengirimkan aspirasi, aspirasi tersebut akan masuk pada baris pertama pada *timeline*.

Gambar 19 Implementasi halaman menu info panen

Gambar 20 Implementasi halaman menu aspirasi

Pengkodean Iterasi Kedua (XP Coding)

Tahap pengkodean iterasi kedua dilakukan dengan memperbaiki REST fungsi *post* informasi hasil panen pada iterasi pertama. Gambar 21 menunjukkan potongan kode REST untuk *post* informasi hasil panen. Potongan kode yang ditandai dengan garis merah pada Gambar 21 digunakan untuk mengupload foto komoditas.

Ketika proses upload foto tidak berhasil REST akan memberikan pesan “*Cannot upload file*”, sedangkan ketika proses upload foto berhasil REST akan menjalankan *query* untuk memasukkan data informasi hasil panen ke dalam *database*. REST akan memberikan pesan “Berhasil mengirim info panen” jika *query* yang dilakukan berhasil. Jika tidak REST akan memberikan pesan “*Error my executing MySQL query*”.

```

router.post("/info_petani", function(req,res){
  upload(req,res,function(err) {
    if(err) {
      res.json({"status" : "error" , "message" : "Cannot
      upload file"});
    }

    var query="INSERT INTO ??(?,?,?,?,,?,?,,?) VALUES
    (?,,?,,?,,?,,?,,?)";
    var table= ["info_petani", "id_komoditas", "id_user",
    "time", "latitude", "longitude", "lokasi",
    "jumlah_komoditas", "keterangan", "foto",
    req.body.id_komoditas, req.body.id_user, new
    Date(req.body.time) ,req.body.latitude, req.body.longitude,
    req.body.lokasi, req.body.jumlah_komoditas,
    req.body.keterangan, req.file.filename];
    query = mysql.format(query,table);
    connection.query(query,function(err,rows){
      if(err){
        res.json({"status" : "error" , "message" : "Error
        executing MySQL query"});
      }else{
        res.json({"status" : "error" , "message" : "Berhasil
        mengirim info panen" });
      }
    });
  });
});

```

Gambar 21 REST *post* informasi hasil panen iterasi kedua

Pada tahap pengkodean iterasi kedua juga dilakukan penambahan token yang bertujuan agar data tidak dapat diakses oleh setiap orang. Pengkodean token dapat dilihat pada Gambar 22. Token yang digunakan masih bersifat statis yaitu dengan menggunakan kata 'portalharga' yang nantinya akan diimplementasikan pada bagian belakang *URL* masing-masing kode REST. Contoh *URL* penggunaan token untuk mengakses REST dapat dilihat pada Gambar 23.

```

// Auth token
router.use(function(req,res,next){
  console.log(req.method,req.url);
  var token = req.body.token || req.query.token ||
req.headers['x-access-token'];
  if(token == 'portalharga'){
    next();
  }else{
    return res.status(403).send({
      "Error": true,
      "Message": 'No token provided.'
    });
  }
});

```

Gambar 22 Potongan kode untuk token

```
http://localhost:3000/api_portalharga/komoditas/?token=portalharga
```

Gambar 23 Contoh penggunaan token

Untuk mengimplementasikan REST dapat menggunakan AJAX Gambar 24 merupakan potongan AJAX untuk fungsi *post* info panen raya. Bagian A pada Gambar 25 menunjukkan *method* REST yang digunakan yaitu *method POST*. *URL* yang ditunjukkan oleh bagian B pada Gambar 24 merupakan *URL REST method post* untuk informasi hasil panen raya. *Form* info panen raya terdapat *field* untuk menampilkan komoditas pertanian sehingga perlu dibuat kode AJAX untuk *get* komoditas yang dapat dilihat pada Gambar 25. *Pair Programming* pada iterasi kedua dilakukan dengan pertemuan rutin dengan tim internal pengembangan web yaitu dengan berdiskusi tentang pengkodean REST untuk fungsi melihat info harga komoditas pertanian.

```
var data = new FormData(this);

$.ajax({
  type: 'POST', A
  url: B
  "http://localhost:3000/api_portalharga/info_petani?token=portalharga",
  data:data,
  cache:false,
  contentType: false,
  processData: false,
  complete: function() {
    setTimeout(function() {
      location.reload();
    }, 2000);
  },
  success: function(msg) {
    // console.log(msg.data);
    if(msg.status == 'success'){

      Materialize.toast(msg.message, 4000);
      return true;
    }
    else {
      Materialize.toast(msg.message, 4000);
      return false;
    }
  },
  error: function(data){
    Materialize.toast(data, 4000);
    return false;
  }
});
```

Gambar 24 Potongan AJAX *post* info panen

```
$.ajax({
  type: 'GET',
  url:
    "http://localhost:3000/api_portalharga/komoditas?token=por
    talharga",
  success: function(msg) {
    $("#komoditas").empty().html(' ');
    $("#komoditas").append($("<option disabled
    selected></option>").val("").html("Pilih Komoditas"));
    jQuery.each( msg.data, function( i, item )
  {
    $("#komoditas").append($("<option></option>").val(item
    .id_komoditas).html(item.nama_komoditas));
  });

  // $("#komoditas").trigger('contentChanged');
  $("#komoditas").material_select();
  }
});
```

Gambar 25 Potongan AJAX *get* komoditas

Pengujian Iterasi Kedua (XP *Testing*)

Tahap pengujian iterasi kedua dilakukan dengan menguji REST untuk fungsi melihat info harga dan melihat info panen raya pada *software* Postman. Hasil pengujian pada iterasi kedua dapat dilihat pada Tabel 5.

Tabel 5 Pengujian fungsi REST iterasi kedua

<i>Actor</i>	Fungsi REST	Status
Petani	<i>Post</i> info Panen	Berhasil
	<i>Post</i> aspirasi	Berhasil
	Melihat <i>timeline</i> aspirasi	Berhasil
	Melihat info harga	Berhasil
	Melihat info panen raya	Berhasil

Gambar 26 merupakan *output* REST dari fungsi melihat info harga komoditas pertanian. *Array* JSON pada Gambar 26 merupakan data harga komoditas pertanian yang terdiri atas dua objek. Objek pertama terdiri 10 atribut. Atribut *id_harga* memiliki *value* 2, hal ini menunjukkan bahwa data harga tersebut memiliki id 2. Atribut *harga* memiliki *value* 3000, hal ini menunjukkan harga dari komoditas yaitu 30000. Atribut *tanggal* memiliki *value* 2016-05-27 20:00:05, hal ini menunjukkan informasi harga komoditas dimasukkan oleh masyarakat pada tanggal 27 Mei 2016. Atribut *id_komoditas* memiliki *value* 2, hal ini menunjukkan komoditas yang dimasukkan memiliki id 2. Atribut *id_user* memiliki *value* 1, hal ini menunjukkan bahwa pengguna yang memasukkan informasi harga komoditas memiliki id 1. Atribut *foto* memiliki *value* *cabai_merah_keriting.jpg*, *value* ini menunjukkan *path* foto yang digunakan. Atribut *id_pasar* dan *nama_pasar*

menunjukkan identitas dari pasar. Atribut jenis dan nama_komoditas menunjukkan jenis komoditas yang dimasukkan pengguna.

Selain itu, objek kedua terdiri 10 atribut. Atribut id_harga memiliki *value* 2, hal ini menunjukkan bahwa data harga tersebut memiliki id 2. Atribut harga memiliki *value* 110000, hal ini menunjukkan harga dari komoditas yaitu 110000. Atribut tanggal memiliki *value* 2016-05-27 20:00:05, hal ini menunjukkan informasi harga komoditas dimasukkan oleh masyarakat pada tanggal 27 Mei 2016. Atribut id_komoditas memiliki *value* 5, hal ini menunjukkan komoditas yang dimasukkan memiliki id 5. Atribut id_user memiliki *value* 7, hal ini menunjukkan bahwa pengguna yang memasukkan informasi harga komoditas memiliki id 71. Atribut foto memiliki *value* bawang_merah.jpg, *value* ini menunjukkan *path* foto yang digunakan. Atribut id_pasar dan nama_pasar menunjukkan identitas dari pasar. Atribut jenis dan nama_komoditas menunjukkan jenis komoditas yang dimasukkan pengguna.

```
{
  "status": "success",
  "message": "success",
  "data": [
    {
      "id_harga": 2,
      "harga": 30000,
      "tanggal": "2016-05-27 20:00:05",
      "id_komoditas": 2,
      "id_user": 1,
      "foto": "cabai_merah_keriting.jpg",
      "id_pasar": 2,
      "nama_pasar": "Pasar Jatinegara",
      "jenis": "Cabai",
      "nama_komoditas": "Cabai Merah Keriting"
    },
    {
      "id_harga": 4,
      "harga": 110000,
      "tanggal": "2016-05-27 20:00:05",
      "id_komoditas": 6,
      "id_user": 7,
      "foto": "bawang_merah.jpg",
      "id_pasar": 2,
      "nama_pasar": "Pasar Jatinegara",
      "jenis": "Cabai",
      "nama_komoditas": "Cabai Merah Keriting"
    }
  ]
}
```

Gambar 26 *Output* REST fungsi melihat info harga

Gambar 27 merupakan *output* dari fungsi *post* info panen raya. *Output* berupa data JSON. Array JSON pada Gambar 27 yaitu *array* data hasil panen yang memiliki satu objek yang terdiri atas 12 atribut. Atribut id_info_petani memiliki *value* 47, hal ini menunjukkan bahwa data info petani yang dimasukkan memiliki id

47. Atribut `id_komoditas` memiliki *value* 5, hal ini menunjukkan bahwa komoditas yang dipanen memiliki id yaitu komoditas cabai rawit hijau. Atribut `id_user` memiliki *value* 9, hal ini menunjukkan bahwa pengguna yang memasukkan hasil panen raya memiliki identitas unik 9. Atribut `time_start` memiliki *value* 2017-01-26 00:00:00.000, hal ini menunjukkan bahwa perkiraan waktu awal pengguna mengalami panen raya yaitu tanggal 26 Januari 2017. Atribut `time_end` memiliki *value* 2017-01-28 00:00:00.000, hal ini menunjukkan bahwa perkiraan waktu awal pengguna mengalami panen raya yaitu tanggal 28 Januari 2017. Atribut `lokasi` memiliki *value* Depok City, West Java, Indonesia, hal ini menunjukkan bahwa lokasi pengguna yang memasukkan info hasil panen raya berada di Kota Depok. Atribut `latitude` dan `longitude` memiliki *value* masing-masing -6.4024844 dan 106.7942405, *value* dari atribut `latitude` dan `longitude` ini menunjukkan koordinat geografis dari Kota Depok. Atribut `jumlah_komoditas` memiliki *value* 26, hal ini menunjukkan bahwa perkiraan jumlah komoditas yang dipanen sebanyak 26 ton. Atribut `keterangan` memiliki *value* cabai rawit hijau masih segar, atribut `keterangan` ini merupakan atribut opsional. Atribut `username` dan `nama` merupakan atribut yang menunjukkan identitas dari pengguna, masing-masing memiliki *value* shellafuribiru dan shellafuri biru mardika.

```
{
  "status": "success",
  "message": "success",
  "data": [
    {
      "id_info_petani": 47,
      "id_komoditas": 5,
      "id_user": 9,
      "time_start": "2017-01-26 00:00:00.000",
      "time_end": "2017-01-28 00:00:00.000",
      "latitude": "-6.4024844",
      "longitude": "106.7942405",
      "lokasi": "Depok Jaya, Indonesia",
      "jumlah_komoditas": 50,
      "keterangan": "cabai rawit hijau masih segar",
      "foto": "cabe rawit hijau.jpg-1485465148022.jpg",
      "nama": "shellafuri biru mardika"
    }
  ]
}
```

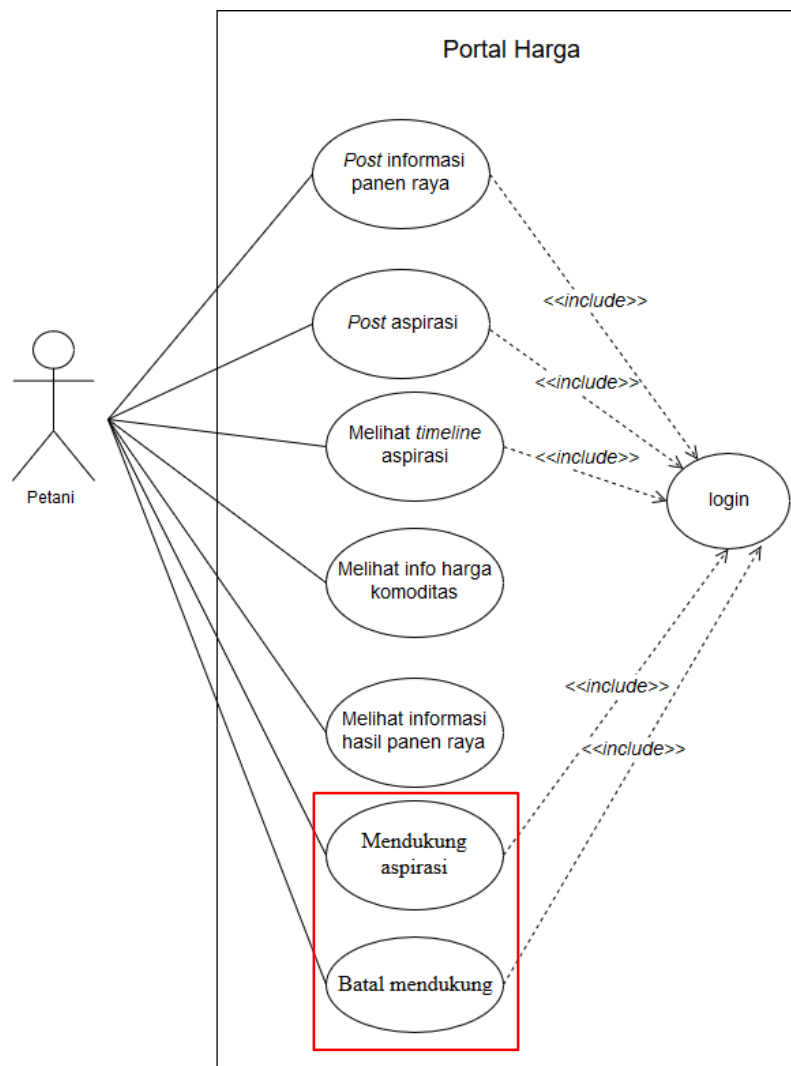
Gambar 27 *Output* REST fungsi melihat informasi hasil panen raya

Iterasi Ketiga

Perencanaan Iterasi Ketiga (XP Planning)

Tahap perencanaan pada iterasi ketiga dilakukan perbaikan *use case* dengan penambahan fungsi dukung apsirasi dan fungsi batal mendukung. Perbaikan *use case* tersebut ditandai dengan garis merah yang dapat dilihat pada Gambar 28.

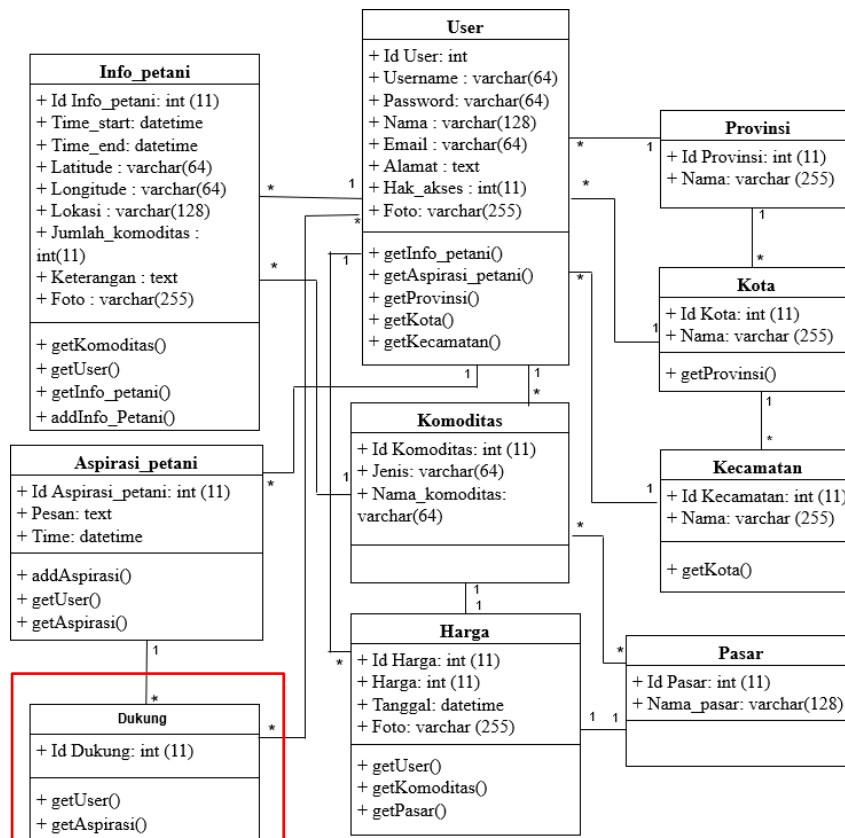
Petani harus melakukan *login* terlebih dahulu agar dapat mendukung aspirasi dan membatalkan dukungan.



Gambar 28 Use case diagram iterasi ketiga

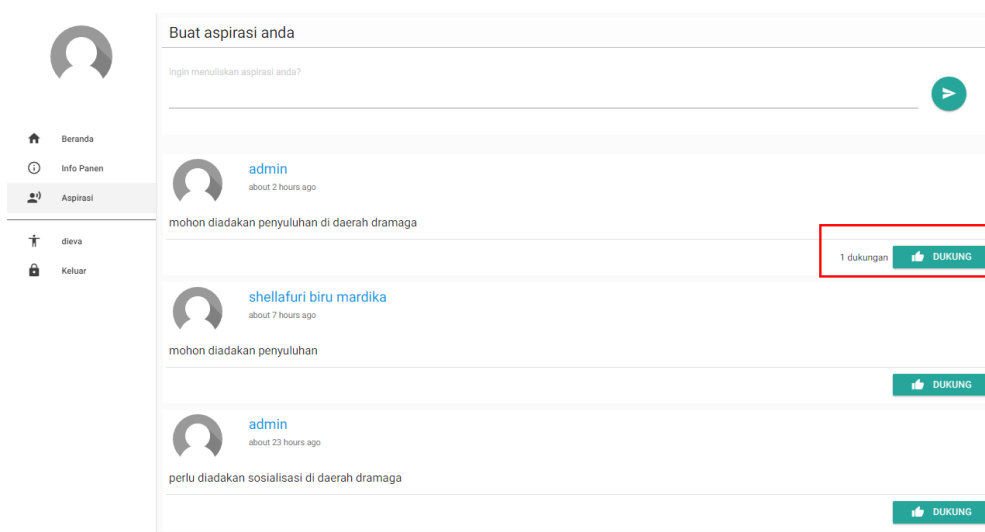
Desain Iterasi Ketiga (XP Design)

Desain iterasi ketiga yaitu melakukan perbaikan *class diagram*. Gambar 29 merupakan *class diagram* pada iterasi ketiga. Perbaikan *class diagram* yaitu dengan penambahan *class* dukung yang ditandai dengan garis merah pada Gambar 29. Tabel dukung terdiri atas atribut id dengan *type* data int (11). *Class* dukung ini akan digunakan untuk menyimpan data pengguna yang mendukung aspirasi. *Class* aspirasi_petani dan *class* dukung memiliki hubungan dengan nilai kardinalitas satu ke banyak, hal ini menunjukkan bahwa satu aspirasi dapat memiliki banyak dukungan. Sementara itu, *class* user dan *class* dukung juga memiliki hubungan dengan nilai kardinalitas satu ke banyak, hal ini menunjukkan bahwa satu *user* dapat memiliki banyak dukungan.

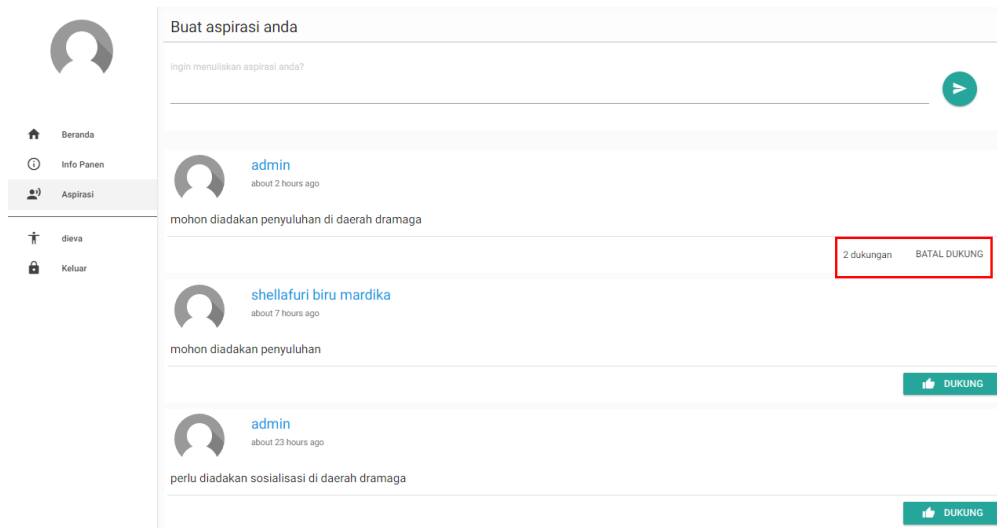


Gambar 29 Class diagram iterasi ketiga

Gambar 30 merupakan implementasi tampilan untuk fungsi mendukung aspirasi pada web. Ketika pengguna menekan *button* dukung, *button* tersebut otomatis akan berubah menjadi *button* batal dukung yang dapat dilihat pada Gambar 31. Ketika pengguna ingin membatalkan dukungan, pengguna dapat menekan *button* batal dukung tersebut. Selain terdapat *button* untuk mendukung, sistem juga menyediakan informasi jumlah petani yang mendukung aspirasi tersebut.



Gambar 30 Implementasi halaman fungsi dukung aspirasi



Gambar 31 Implementasi halaman fungsi batal dukung aspirasi

Pengkodean Iterasi Ketiga (XP Coding)

Pengkodean iterasi ketiga dilakukan dengan membuat REST untuk fungsi dukung aspirasi yang dapat dilihat pada Gambar 32. *Method* yang digunakan untuk fungsi dukung aspirasi yaitu *method post*. Bagian yang ditandai oleh garis merah pada Gambar 32 digunakan untuk melakukan *query* ke *database*. Bagian A pada Gambar 32 digunakan untuk atribut-atribut yang ada pada tabel dukung. Bagian B pada Gambar 32 menunjukkan nilai dari atribut yang didapat dari *request query*. Jika proses *query* gagal maka *server* akan memberikan pesan “Error executing MySQL query”. Proses *query* berhasil maka *server* akan memberikan pesan “Berhasil mendukung aspirasi”.

```
router.post("/dukung", function(req, res) {
  var query="INSERT INTO ??(??,??) VALUES (??,??)";
  var table= ["dukung", "id_aspirasi", "id_user",
    req.query.id_aspirasi, req.query.id_user];
  query = mysql.format(query,table);

  connection.query(query,function(err,rows) {
    if (err){
      res.json({"status" : "error" , "message" : "Error
        executing MySQL query"});
    }
    else {
      res.json({"status" : "success", "message" : "Berhasil
        mendukung aspirasi"});
    }
  });
});
```

Gambar 32 Potongan REST *post* dukung aspirasi

Gambar 33 merupakan kode REST untuk mendapatkan data pengguna yang mendukung aspirasi. Bagian yang ditandai garis merah pada Gambar 33 digunakan

untuk melakukan proses *query*. *Query* digunakan yaitu SELECT. Jika proses *query* gagal, *server* akan menampilkan pesan “*Error executing MySQL query*”. Jika proses *query* berhasil *server* akan menampilkan pesan “*success*” dan akan menampilkan data.

```
router.get("/dukung/:id_aspirasi",function(req,res){
    var query = "SELECT * FROM dukung WHERE id_aspirasi = " +
        req.params.id_aspirasi;
    query = mysql.format(query);

    connection.query(query,function(err,rows){
        if (err){
            res.json({"status" : "error" , "message" : "Error
            executing MySQL query"});
        } else {
            res.json({"status" : "success", "message" : "Success",
            "data" : rows});
        }
    });
});
```

Gambar 33 Potongan REST *get* dukung aspirasi

Gambar 34 menunjukkan *output* REST dari *get* dukung aspirasi. *Array* pada Gambar 34 merupakan data dukung aspirasi yang memiliki dua objek. Objek pertama terdiri dari 3 atribut. Atribut *id* memiliki *value* 2. Atribut *id_aspirasi* memiliki *value* 51, hal ini menunjukkan bahwa aspirasi yang didukung memiliki id 51. Atribut *id_user* memiliki *value* 8, hal ini menunjukkan bahwa pengguna yang mendukung aspirasi memiliki id 8.

Selain itu, objek kedua terdiri dari 3 atribut. Atribut *id* memiliki *value* 3, Atribut *id_aspirasi* memiliki *value* 51, hal ini menunjukkan bahwa aspirasi yang didukung memiliki id 51. Atribut *id_user* memiliki *value* 8, hal ini menunjukkan bahwa pengguna yang mendukung aspirasi memiliki id 9.

```
{
  "status": "success",
  "message": "Success",
  "data": [
    {
      "id": 2,
      "id_aspirasi": 51,
      "id_user": 8
    },
    {
      "id": 3,
      "id_aspirasi": 51,
      "id_user": 9
    }
  ]
}
```

Gambar 34 *Output* dari REST *get* dukung

Potongan AJAX mendukung aspirasi dapat dilihat pada Gambar 35. Pada Gambar 35 potongan AJAX yang utama berada pada bagian 1 yaitu berfungsi ketika pengguna menekan *button* dukung, kode tersebut akan mengirim data ke tabel dukung pada *database*. Ketika pengiriman data ini berhasil *button* dukung akan berubah menjadi *button* batal dukung dan akan melakukan fungsi pada Gambar 35. Bagian 2 yaitu kode untuk mengambil data dukung aspirasi yang akan digunakan untuk menghitung jumlah pengguna yang mendukung aspirasi tersebut. *Pair programming* pada iterasi ketiga yaitu hanya melakukan diskusi dengan tim pengembangan melalui perantara percakapan grup pada media sosial. Potongan kode REST untuk batal dukungan dapat dilihat pada Lampiran 4.



Gambar 35 Potongan AJAX untuk mendukung aspirasi

Pengujian Iterasi Ketiga (XP Testing)

Tahap pengujian iterasi kedua dilakukan dengan menguji REST API yang telah dibuat beserta implementasi REST API ke dalam aplikasi web. Pengujian dilakukan oleh tim secara internal menggunakan *software* Postman. Hasil pengujian REST API dapat dilihat pada Tabel 6.

Tabel 6 Pengujian REST iterasi ketiga

<i>Actor</i>	Fungsi REST	Status
Petani	<i>Post</i> info Panen	Berhasil
	<i>Post</i> aspirasi	Berhasil
	Melihat <i>timeline</i> aspirasi	Berhasil
	Melihat info harga	Berhasil
	Melihat info panen raya	Berhasil
	Mendukung aspirasi	Berhasil

SIMPULAN DAN SARAN

Simpulan

Penelitian ini telah berhasil mengembangkan web berbasis REST API dalam aplikasi web Portal Harga untuk petani. Aplikasi Portal Harga dibuat dengan menggunakan metode *Extreme Programming* (XP) sebanyak tiga iterasi. Aplikasi Portal Harga digunakan untuk membantu petani menyampaikan informasi hasil panen raya, menyampaikan aspirasi kepada pemerintah, melihat informasi harga komoditas pertanian, melihat informasi panen raya, dan mendukung suatu aspirasi. Hasil pengujian dengan menggunakan metode *black box* menunjukkan bahwa fungsi utama aplikasi Portal Harga telah bekerja dengan baik sesuai dengan kebutuhan aplikasi.

Saran

Penelitian selanjutnya dapat mengembangkan aplikasi portal harga dengan menggunakan token Oauth dinamis agar lebih terjamin keamanannya. Selain itu, penelitian selanjutnya dapat melakukan integrasi *login* dengan media sosial seperti facebook atau google+ sehingga dapat memudahkan petani mendaftarkan diri ke aplikasi Portal Harga.

DAFTAR PUSTAKA

- Fauziah Y. 2013. Aplikasi iklan baris online menggunakan arsitektur REST web service. *Telematika*. 9(2):75-80
- Grihadevi K. 2016. Pengembangan Modul Pelaporan Harga Komoditas Pertanian pada Aplikasi Portal Harga untuk Masyarakat menggunakan REST API. [skripsi]. Bogor (ID): Institut Pertanian Bogor.
- Hidayat SG, Widodo AP. 2012. Pembuatan prototipe perangkat lunak perhitungan investasi agribisnis hortikultura pada off season. SNASTI. Sekolah Tinggi Manajemen Informatika & Teknik Komputer, Surabaya.
- [IPJ] Info Pangan Jakarta. 2017. Statistik Harga Pangan [internet]. [diunduh 2017 Januari 2]. Tersedia pada : http://infopangan.jakarta.go.id/publik/report_commodity.

- [JSON] JavaScript Object Notation. 2002. Introduction JSON [internet]. [diunduh 2016 Des 23]. Tersedia pada: <http://json.org/json-id.html>.
- Kurniawan E. 2014. Implementasi REST Web Service untuk Sales Order dan Sales Tracking Berbasis Mobile. *Jurnal EKSIS*. 7(1):(1-12).
- Mardan A. 2016. *Pro Express.JS*. New York (US): Apress.
- Mohammadi S, Nikkahan B, Sohrabi S. 2009. Challenges of *user involvement* in extreme programming projects. *International Journal of Software Engineering and Its Applications*. 3(1): 20.
- Pressman RS. 2010. *Software Engineering : A Practitioner's Approach*. Ed ke-7. Boston(US): Mc Graw Hill.
- [RI] Republik Indonesia. 2013. Undang-Undang Republik Indonesia Nomor 19 Tahun 2013 tentang Perlindungan dan Pemberdayaan Petani. Jakarta (ID): Sekretariat Negara
- [Sekjen Kemenperin] Sekretariat Jenderal Kementerian Pertanian. 2013. Statistik Ketenagakerjaan Sektor Pertanian Tahun 2013. Jakarta (ID): Pusat Data dan Sistem Informasi Pertanian
- Setiana D. 2016. Pengembangan Modul Pelaporan Harga Komoditas Pertanian pada Aplikasi Portal Harga untuk Masyarakat menggunakan REST API. [skripsi]. Bogor (ID): Institut Pertanian Bogor.
- Setyatama D. 2016. Pengembangan Aplikasi Pelaporan Harga Komoditas Pertanian Berbasis Mobile menggunakan REST API. [skripsi]. Bogor (ID): Institut Pertanian Bogor.
- [XP] Extreme Programming. 2016. The Rules of Extreme Programming. [internet] [diunduh pada 2016 Desember 23]: Tersedia pada <http://www.extremeprogramming.org/rules.html>

Lampiran 1 *Fully developed use case description* aplikasi Portal HargaA) *Fully developed use case description* fungsi *post aspirasi*

<i>Use case</i>	Deskripsi								
<i>Name</i>	<i>Post aspirasi</i>								
<i>Scenario</i>	Petani ingin mengirimkan aspirasi baik berupa kritik maupun saran kepada pemerintah								
<i>Triggering Event</i>	Petani akan memasukkan aspirasi kepada pemerintah								
<i>Brief Description</i>	Petani ingin menyampaikan aspirasi kepada pemerintah. Petani menggunakan aplikasi portal harga untuk mengirimkan aspirasi								
<i>Actors</i>	Petani								
<i>Related Use Case</i>	Login								
<i>Stakeholders</i>	Petani								
<i>Preconditions</i>	Petani telah berhasil melakukan <i>login</i> dan memilih menu aspirasi								
<i>Post conditions</i>	Petani berhasil mengirimkan data aspirasi								
<i>Flow of Events</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 menu aspirasi</td></tr> <tr> <td>2 Mengisi <i>form</i> aspirasi</td><td></td></tr> <tr> <td>3 Menekan tombol kirim</td><td>3.1 Menyimpan data input</td></tr> </table>	<i>Actor</i>	<i>System</i>	1 Memilih menu aspirasi	1.1 Menampilkan halaman menu aspirasi	2 Mengisi <i>form</i> aspirasi		3 Menekan tombol kirim	3.1 Menyimpan data input
<i>Actor</i>	<i>System</i>								
1 Memilih menu aspirasi	1.1 Menampilkan halaman menu aspirasi								
2 Mengisi <i>form</i> aspirasi									
3 Menekan tombol kirim	3.1 Menyimpan data input								
<i>Exceptions Conditions</i>	<ul style="list-style-type: none"> - Jika petani tidak mengisi <i>form</i> secara lengkap, petani tidak bisa mengirimkan aspirasi - Jika petani belum melakukan <i>login</i>, petani tidak bisa mengirimkan aspirasi 								

B) *Fully developed use case description* melihat *timeline* aspirasi

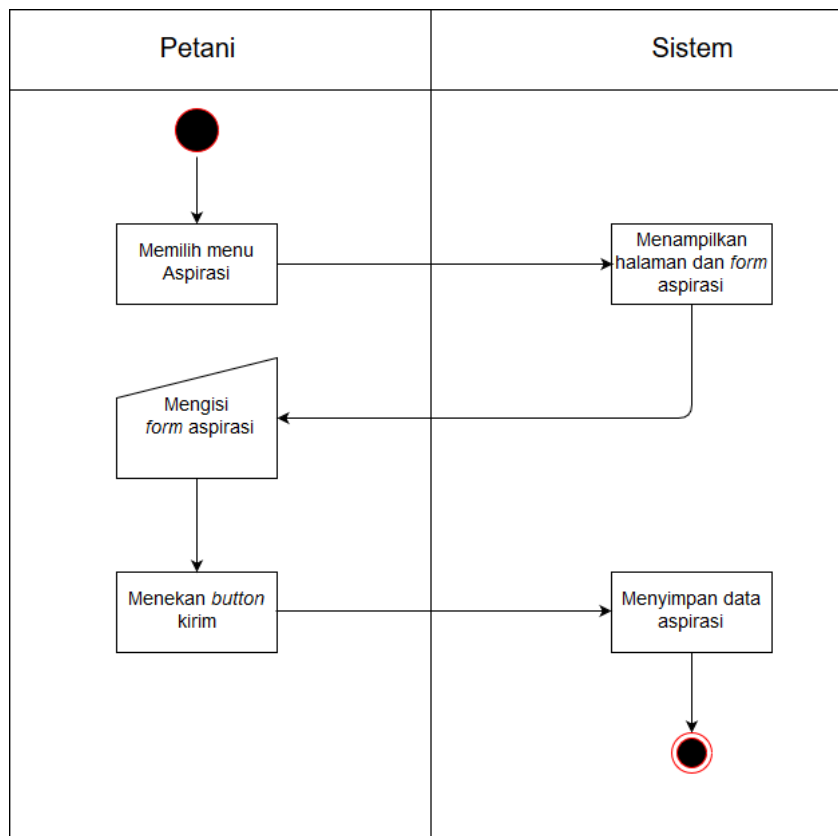
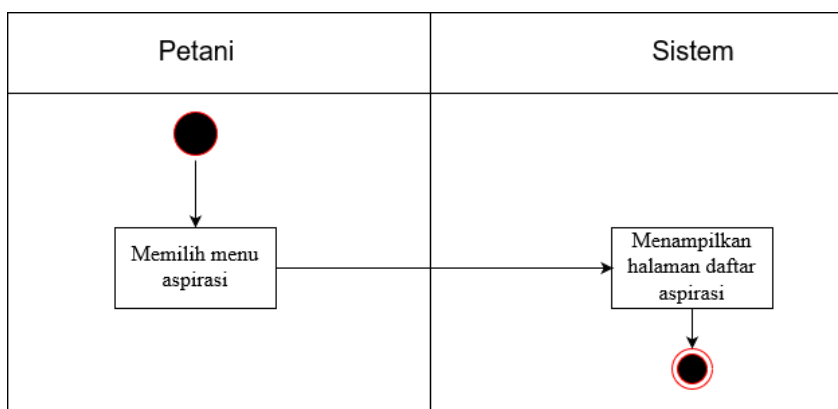
<i>Use case</i>	Deskripsi				
<i>Name</i>	Melihat <i>timeline</i> aspirasi				
<i>Scenario</i>	Petani ingin melihat <i>timeline</i> aspirasi				
<i>Triggering Event</i>	Petani melihat <i>timeline</i> aspirasi				
<i>Brief Description</i>	Petani melihat aspirasi yang telah dimasukkan oleh petani				
<i>Actors</i>	Petani				
<i>Related Use Case</i>	Login				
<i>Stakeholders</i>	Petani				
<i>Preconditions</i>	Petani telah berhasil melakukan <i>login</i> dan memilih menu aspirasi				
<i>Post conditions</i>	Petani melihat aspirasi				
<i>Flow of Events</i>	<table> <tr> <th><i>Actor</i></th><th><i>System</i></th></tr> <tr> <td>Memilih menu aspirasi</td><td>Menampilkan halaman menu aspirasi</td></tr> </table>	<i>Actor</i>	<i>System</i>	Memilih menu aspirasi	Menampilkan halaman menu aspirasi
<i>Actor</i>	<i>System</i>				
Memilih menu aspirasi	Menampilkan halaman menu aspirasi				
<i>Exceptions Conditions</i>	Jika belum ada petani yang memasukkan aspirasi maka tidak ada data yang ditampilkan				

Lampiran 1 *Fully developed use case description* aplikasi Portal Harga (Lanjutan)C) *Fully developed use case description* untuk fungsi melihat info harga komoditas pertanian

<i>Use case</i>	Deskripsi				
<i>Name</i>	Melihat info harga komoditas pertanian				
<i>Scenario</i>	Petani ingin melihat info harga komoditas pertanian				
<i>Triggering Event</i>	Petani akan melihat info harga komoditas pertanian				
<i>Brief Description</i>	Melihat info harga komoditas pertanian yang telah dimasukkan oleh masyarakat				
<i>Actors</i>	Petani				
<i>Related Use Case</i>	Login				
<i>Stakeholders</i>	Petani				
<i>Preconditions</i>	Petani telah berhasil melakukan <i>login</i> dan memilih menu beranda				
<i>Post conditions</i>	Petani melihat info harga komoditas pertanian				
<i>Flow of Events</i>	<table border="0"> <tr> <td><i>Actor</i></td><td><i>System</i></td></tr> <tr> <td>Memilih menu beranda</td><td>Menampilkan info harga komoditas pertanian</td></tr> </table>	<i>Actor</i>	<i>System</i>	Memilih menu beranda	Menampilkan info harga komoditas pertanian
<i>Actor</i>	<i>System</i>				
Memilih menu beranda	Menampilkan info harga komoditas pertanian				
<i>Exceptions Conditions</i>	Jika belum ada masyarakat yang memasukkan harga komoditas pertanian maka tidak ada data yang ditampilkan				

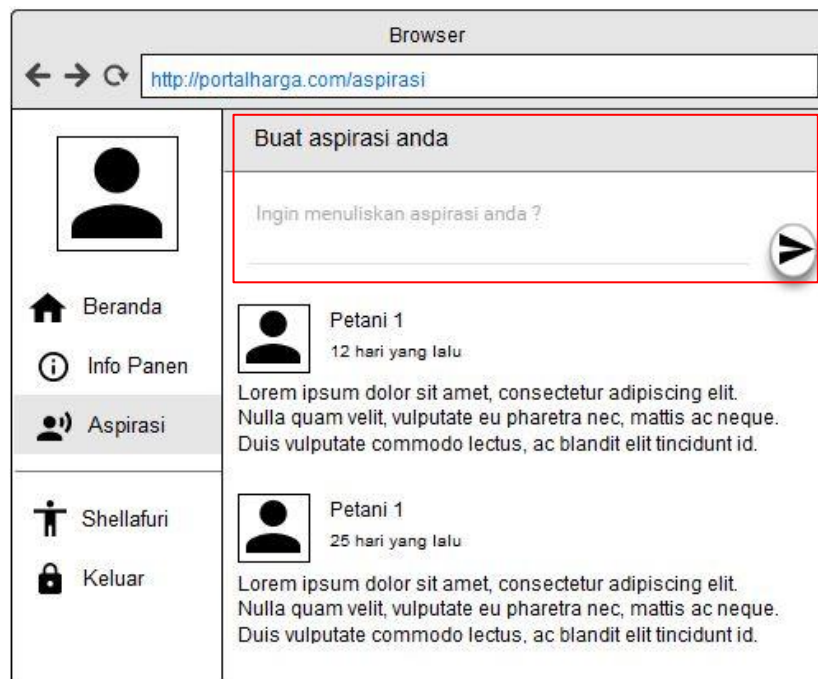
D) *Fully developed use case description* untuk fungsi melihat informasi hasil panen raya

<i>Use case</i>	Deskripsi				
<i>Name</i>	Melihat informasi hasil panen raya				
<i>Scenario</i>	Petani ingin melihat informasi hasil panen raya				
<i>Triggering Event</i>	Petani akan melihat informasi hasil panen raya				
<i>Brief Description</i>	Petani menggunakan aplikasi portal harga untuk melihat informasi hasil panen raya yang dimasukkan oleh petani				
<i>Actors</i>	Petani				
<i>Related Use Case</i>	Login				
<i>Stakeholders</i>	Petani				
<i>Preconditions</i>	Petani telah berhasil melakukan <i>login</i> dan memilih menu beranda				
<i>Post conditions</i>	Petani melihat info harga komoditas pertanian				
<i>Flow of Events</i>	<table border="0"> <tr> <td><i>Actor</i></td><td><i>System</i></td></tr> <tr> <td>Memilih menu beranda</td><td>Menampilkan informasi hasil panen raya</td></tr> </table>	<i>Actor</i>	<i>System</i>	Memilih menu beranda	Menampilkan informasi hasil panen raya
<i>Actor</i>	<i>System</i>				
Memilih menu beranda	Menampilkan informasi hasil panen raya				
<i>Exceptions Conditions</i>	Jika belum ada petani yang memasukan data hasil panen raya maka tidak ada data yang ditampilkan				

Lampiran 2 *Swimlane* aplikasi Portal HargaA) *Swimlane* untuk fungsi *post* aspirasiB) *Swimlane* untuk fungsi melihat *timeline* aspirasi

Lampiran 3 *Mockup* halaman aplikasi Portal Harga

A) *Mockup* halaman *post* aspirasi



B) *Mockup* halaman melihat *timeline* aspirasi



Lampiran 4 Potongan kode REST

A) *Update* informasi hasil panen raya

```

router.put("/info_petani/:info_petani", function(req, res) {
  upload(req, res, function(err) {
    if (err) {
      res.json({
        "status": "error",
        "message": "Cannot upload file"
      });
    }
    if (req.file) {
      var query = "UPDATE ?? SET ?? = ?, ?? = ?, ?? = ?, ?? = ?,
        ?? = ?, ?? = ?, ?? = ?, ?? = ?, ?? = ?, ?? = ? WHERE ?? =
        ?";
      var table = ["info_petani", "id_komoditas",
        req.body.id_komoditas, "id_user", req.body.id_user,
        "time_start", new Date(req.body.time_start), "time_end",
        new Date(req.body.time_start), "latitude",
        req.body.latitude, "longitude", req.body.longitude,
        "lokasi", req.body.lokasi,
        "jumlah_komoditas", req.body.jumlah_komoditas,
        "keterangan", req.body.keterangan,
        "foto", req.file.filename, "id_info_petani",
        req.params.info_petani];
      query = mysql.format(query, table);
      connection.query(query, function(err, rows) {
        if (err) {
          res.json({
            "status": "error",
          });
        } else {
          var query = "SELECT * FROM ?? WHERE ??=?";
          var table = ["info_petani", "id_info_petani",
            req.params.info_petani];
          query = mysql.format(query, table);
          connection.query(query, function(err, rows) {
            if (err) {
              res.json({
                "status": "error",
                "message": "Error executing MySQL query"
              });
            }
          });
        }
      });
    }
  });
});

```

Lampiran 4 Potongan kode REST (Lanjutan)

```

    } else {
        res.json({
            "status": "success",
            "message": "Update info panen berhasil",
            "data": rows
        });
    }
} else {
    var query = "UPDATE ?? SET ?? = ?, ?? = ?, ?? = ?, ?? = ?,
    ?? = ?, ?? = ?, ?? = ?, ?? = ?, ?? = ? WHERE ?? = ?";
    var table = ["info_petani", "id_komoditas",
    req.body.id_komoditas, "id_user", req.body.id_user,
    "time_start", new Date(req.body.time_start), "time_end",
    new Date(req.body.time_start), "latitude", req.body.latitude,
    "longitude", req.body.longitude, "lokasi", req.body.lokasi,
    "jumlah_komoditas", req.body.jumlah_komoditas, "keterangan",
    req.body.keterangan, "id_info_petani", req.params.info_petani];
    query = mysql.format(query, table);
    connection.query(query, function(err, rows) {
        if (err) {
            res.json({
                "status": "error",
                "message": "Error executing MySQL query"
            });
        } else {
            var query = "SELECT * FROM ?? WHERE ??=?";
            var table = ["info_petani", "id_info_petani",
            req.params.info_petani];
            query = mysql.format(query, table);
            connection.query(query, function(err, rows) {
                if (err) {
                    res.json({
                        "status": "error",
                        "message": "Error executing MySQL query"
                    });
                } else {
                    res.json({
                        "status": "success",
                        "message": "Update info panen berhasil",
                        "data": rows
                    });
                }
            });
        }
    });
}
}

```

Lampiran 4 Potongan kode REST (Lanjutan)

B) *Delete* informasi hasil panen raya

```
router.delete("/info_petani/:info_petani", function(req, res) {
  var query = "DELETE from ?? WHERE ??=?";
  var table = ["info_petani", "id_info_petani",
    req.params.info_petani];
  query = mysql.format(query, table);
  connection.query(query, function(err, rows) {
    if (err) {
      res.json({
        "status": "error",
        "message": "Error executing MySQL query"
      }); } else {
      res.json({
        "status": "success",
        "message": "Anda berhasil menghapus info panen"
      }); }
    });
  });
});
```

C) Batal dukung aspirasi

```
router.delete("/dukung", function(req, res){
  var query = "SELECT * FROM dukung WHERE id_aspirasi = " +
    req.query.id_aspirasi + " AND id_user = " +
    req.query.id_user;
  query = mysql.format(query);
  connection.query(query, function(err, rows){
    if (err){
      res.json({"status" : "error" , "message" : "Error
        executing MySQL query"});
    } else {
      if(rows[0]) {
        var query = "POST from ?? WHERE ??=? AND ??=?";
        var table = ["dukung", "id_aspirasi"
          , req.query.id_aspirasi, "id_user" , req.query.id_user];
        query = mysql.format(query, table);
        connection.query(query, function(err, rows) {
          if(err) {
            res.json({"status" : "error" , "message" : "Error
              executing MySQL query"}); } else {
            res.json({"status" : "success", "message" : "Anda
              berhasil membatalkan dukungan aspirasi"});
          }
        }); } else {
        res.json({"status" : "error", "message" : "Anda tidak
          mendukung aspirasi ini"});
        }
      }
    });
  });
});
```

RIWAYAT HIDUP

Penulis dilahirkan di Brebes, Jawa Tengah, pada tanggal 10 Agustus 1994 sebagai anak kedua dari dua bersaudara dari pasangan Bapak Sumardi dan Ibu Holipah. Penulis lulus dari SMA Negeri 1 Bumiayu pada tahun 2012 dan melanjutkan pendidikan di Departemen Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Pertanian Bogor melalui jalur SNMPTN Undangan.

Selama menjalani perkuliahan, penulis aktif dalam kegiatan akademik dan non-akademik. Penulis aktif dalam organisasi HIMALKOM (Himpunan Mahasiswa Ilmu Komputer) pada tahun 2013-2014. Penulis juga berperan di beberapa kepanitiaan, diantaranya Olimpiade Asrama (OA), IT-Today, Pesta Sains Nasional (PSN). Penulis menerima Beasiswa Prestasi Peningkatan Akademik (PPA) periode tahun 2013-2016. Penulis melaksanakan PKL di PT. Epson Indonesia Jakarta pada bulan Juli-Agustus 2016.