

PENGEMBANGAN MODUL PELAPORAN HARGA PADA APLIKASI DIGITAL TANI

DEVELOPMENT OF PRICE REPORT MODULES ON DIGITAL TANI APPLICATIONS

NUGROHO AGUNG CAHYONO (G64130039)*, DEAN APRIANA RAMADHAN

ABSTRAK

Informasi yang sulit tersampaikan dari petani kepada pemerintah merupakan permasalahan yang terjadi dalam mengelola dan mengembangkan sektor pertanian. Informasi yang sulit tersampaikan tersebut menyebabkan harga jual hasil panen petani di bawah harga yang ditetapkan oleh pemerintah. Perkembangan Internet dapat dijadikan sebagai peluang untuk mempermudah penyampaian informasi dengan menyediakan media yang mampu menghubungkan penyuluh, petani, masyarakat, pedagang, dan pemerintah. Penelitian ini bertujuan untuk memberikan informasi dan media komunikasi antara pemerintah, petani, masyarakat, dan pedagang. Penelitian ini mengembangkan API menggunakan *Framework Express.js* dan basis data non relasional MongoDB. Pengembangan API menggunakan metode *extreme programming* yang merupakan bagian dari *agile software engineering* yang terbagi menjadi tiga iterasi. API yang dikembangkan adalah modul pelaporan harga pada sisi *back end* dengan keluaran dalam bentuk JSON. JSON memiliki format yang terdiri atas status, data, *message*, dan token. API yang dikembangkan dapat diakses oleh *front end* berbasis *mobile* maupun web.

Kata kunci: *extreme programming*, modul pelaporan harga, REST API.

ABSTRACT

The difficulty for delivering an information from farmers to government is a problem that occurs in managing and developing the agricultural sector. It causes the selling price of farmers harvest becomes lower than the price that had been set by the government. The growth of the Internet can be used as an opportunity to facilitate the delivery of information by providing a media which is able to connect agricultural counselor, farmers, society, traders, and government. The purpose of this research is to provide information and communication media between government, farmers, communities, and traders. This API development used Express.js Framework and non relational database (MongoDB). Development of API used extreme programming method that was part of agile software engineering with three iterations. The research developed API for price reporting module on back end side with output in JSON form. JSON had a format consisting of status, data, message, and token. APIs could be accessed by mobile or web application.

Keywords: extreme programming, price reporting module, REST API.

PENDAHULUAN

Latar Belakang

Indonesia adalah negara agraris yang berarti bahwa sektor pertanian sebagai sumber mata pencaharian maupun sebagai penopang pembangunan. Sektor pertanian memiliki daya serap tenaga kerja yang tinggi. Sektor pertanian menyerap angkatan kerja sebanyak 35.76 juta jiwa atau 30.2% dari keseluruhan angkatan kerja nasional pada tahun 2014. Rata-rata kontribusi pertanian terhadap PDB mencapai 10.26% dengan pertumbuhan 3.90% selama periode 2010-2014 (Kementan 2014). Pertanian menjadi sektor yang penting di Indonesia karena menjadi dasar untuk penyediaan sandang, papan, dan pangan.

Informasi yang sulit tersampaikan dari petani kepada pemerintah merupakan permasalahan yang terjadi dalam mengelola dan mengembangkan sektor pertanian. Informasi yang sulit tersampaikan tersebut menyebabkan harga jual hasil panen petani di bawah harga yang ditetapkan oleh pemerintah. Tengkulak memanfaatkan kondisi tersebut dengan membeli hasil panen dari petani dengan harga rendah (Romadhan 2009).

Pesatnya perkembangan teknologi informasi dapat digunakan untuk mengatasi masalah yang terjadi di sektor pertanian. Teknologi informasi yang dapat dikembangkan salah satunya, yaitu media Internet atau komunikasi dunia maya (Elian *et al.* 2014). Berdasarkan survey APJII, pengguna Internet di Indonesia sebanyak 88.1 juta jiwa dari total 254.2 juta jiwa dengan tingkat penetrasi 34.9%. Berdasarkan perangkat yang digunakan untuk mengakses Internet, jumlah terbanyak yaitu pengguna Internet dengan menggunakan telepon seluler, laptop atau *notebook*, *personal computer*, dan tablet (APJII 2016).

Penelitian ini melakukan *reengineering* penelitian yang dilakukan oleh Setyatama (2016), yaitu pengembangan aplikasi pelaporan harga komoditas pertanian berbasis *mobile* menggunakan REST API. Penelitian tersebut perlu pengembangan token yang dinamis untuk keamanan data (Setyatama 2016). Selain itu penelitian yang dilakukan oleh Mardika (2016) tentang pengembangan REST API modul produsen komoditas pertanian pada aplikasi portal harga perlu pengembangan *email* untuk mengatasi pengguna yang lupa *password*. *Reengineering* dilakukan pada bagian token dan basis data relasional MySQL. Token yang digunakan masih bersifat statis dan diubah menjadi token yang dinamis menggunakan *Json Web Token* (JWT). Basis data yang digunakan dalam penelitian ini adalah basis data non relasional MongoDB. Penelitian ini mengembangkan modul pelaporan harga dengan metode *extreme programming* yang merupakan bagian dari *agile software engineering*. Modul yang dikembangkan diharapkan menjadi media informasi antara *stakeholder* terkait dan mampu menyediakan informasi seputar harga komoditas pertanian kepada *stakeholder* terkait secara *real time*.

Perumusan Masalah

Perumusan masalah pada penelitian ini yaitu, bagaimana cara mengembangkan modul pelaporan harga komoditas pertanian dengan arsitektur REST API, sehingga API dapat diakses oleh *platform mobile* atau web.

Tujuan Penelitian

Penelitian ini bertujuan mengembangkan modul pelaporan harga komoditas pertanian menggunakan arsitektur REST API dan basis data non relasional pada sisi *back end*.

Manfaat Penelitian

Manfaat dari penelitian ini adalah:

1. Dari sisi pedagang

- a Informasi harga komoditas pertanian bisa diketahui secara *real time*.
 - b Memasarkan dagangan.
2. Dari sisi masyarakat
 - a Masyarakat mengetahui harga yang beredar di pasar.
 - b Masyarakat ikut berperan dalam pengendalian harga komoditas pertanian dengan melaporkan harga di pasar secara langsung.
3. Dari sisi pemerintah
 - a Membuat kebijakan berdasarkan harga komoditas pertanian di pasar.
 - b Mengetahui daerah yang membutuhkan operasi pasar.

Ruang Lingkup Penelitian

Ruang lingkup dari penelitian ini adalah:

1. Sistem tidak melakukan validasi untuk setiap data yang dimasukan oleh pengguna.
2. Komoditas pertanian yang digunakan sama dengan penelitian sebelumnya yaitu terdiri atas beras, daging sapi, cabai, kedelai, dan bawang merah.

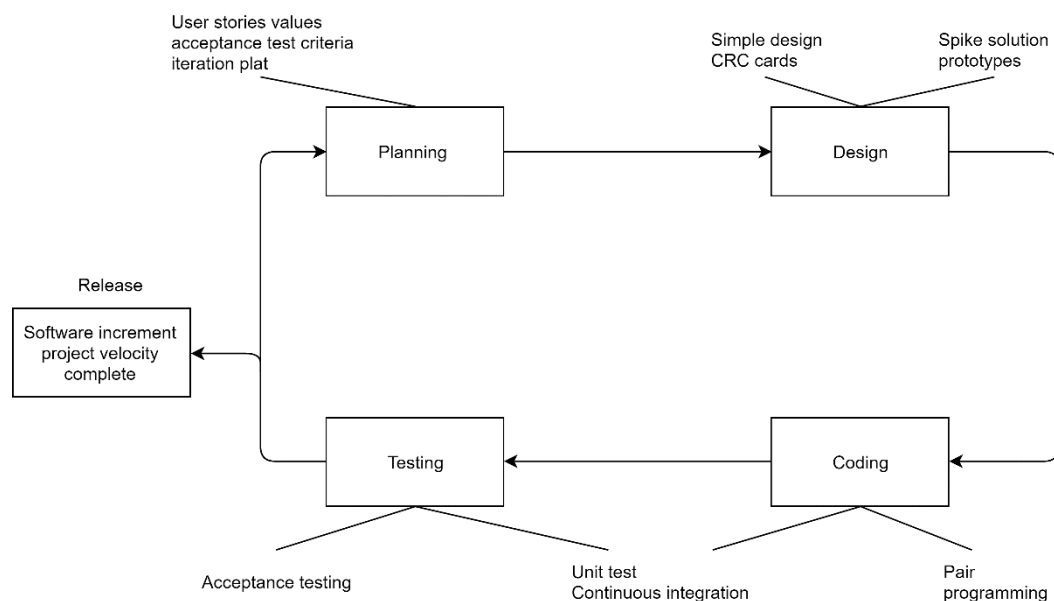
METODE

Data Penelitian

Data yang digunakan antara lain, yaitu komoditas, laporan harga, operasi pasar, dan lokasi. Data tersebut merupakan data dari penelitian sebelumnya yang dilakukan oleh Setyatama (2016). Data lokasi diunduh dari <https://github.com/cahyadsn/daerah>. Data lokasi terdiri dari data provinsi, kabupaten, kecamatan, dan kelurahan, dan jenis.

Tahapan Penelitian

Metode yang digunakan adalah *extreme programming* yang merupakan bagian dari metode *agile software development* yang berbasis pada pengembangan iteratif dan kolaborasi antar anggota tim yang terorganisir. *Extreme programming* merupakan metodologi pengembangan perangkat lunak yang tanggap terhadap perubahan kebutuhan pengguna, sehingga meningkatkan kualitas perangkat lunak (Pressman 2010). Tahapan dalam *extreme programming*, yaitu *planning*, *design*, *coding*, dan *testing* dapat dilihat pada Gambar 1.



Gambar 1 Tahapan pada *Extreme Programming* (Pressman 2010)

1 Perencanaan (*Planning*)

Iterasi yang akan dilakukan selama penelitian sebanyak tiga kali. Tahap perencanaan dimulai dengan tim menentukan *requirements* yang diperlukan dalam pengembangan sistem. *Requirements* yang sudah ditentukan menjadi dasar untuk membuat *user story*, *use case diagram*, modul, dan fungsi. *Acceptance test criteria* dibuat untuk masing-masing modul dan fungsi yang direncanakan.

2 Desain (*Design*)

Tahap desain dimulai dengan membuat *class*, *responsibilities*, dan *collaboration* (CRC) *cards*. *Class* yang sudah dirancang lalu ditentukan atribut beserta dengan fungsi-fungsinya. *Sequence diagram* untuk menggambarkan interaksi dan komunikasi di antara objek dari *class* berdasarkan urutan waktu. Jika mengalami kesulitan *spike solution prototype* dilakukan untuk meminimalkan risiko selama proses pengembangan (Pressman 2010).

3 Pengodean (*Coding*)

Tahap pengodean melakukan *pair programming* dengan anggota tim *back end*, yaitu Fiqih Nur Ramadhan untuk mendukung *continuous integration*. GitHub sebagai media untuk mendukung *pair programming* dan penggabungan *code* pada tim *back end*. Repositori GitHub yang digunakan, yaitu <https://github.com/ryanbaskara/backendPH>.

4 Pengujian (*Testing*)

Pengujian menggunakan metode *black-box* dan dilakukan secara internal oleh tim. Hasil pengujian kemudian diintegrasikan dengan anggota tim *front end*, yaitu Ryan Baskara dan Irfan Rafii. Pengujian sesuai dengan *acceptance test criteria* untuk setiap modul dan fungsi yang diperoleh berdasarkan *user story* pada proses perencanaan (Pressman 2010). Pengujian dilakukan menggunakan aplikasi Postman dengan perintah HTTP (*Hypertext Transfer Protocol*) *method* POST dan GET.

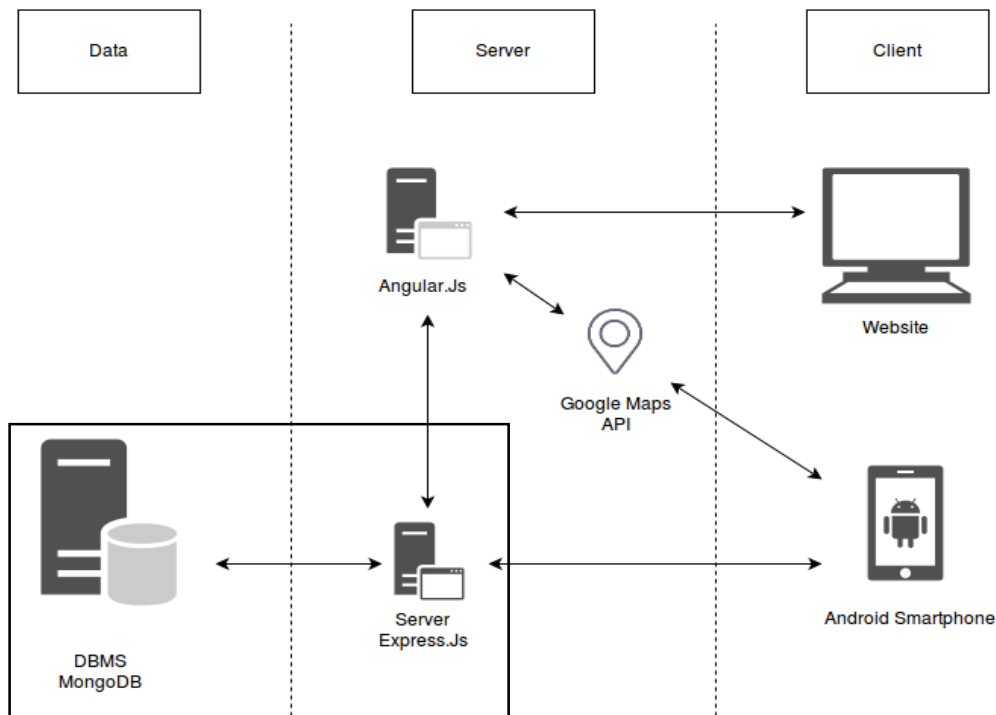
Lingkungan Pengembangan

Spesifikasi perangkat keras yang digunakan pada penelitian ini adalah laptop dengan spesifikasi *Processor* Intel® Core™ i3-3217u, RAM 4 GB, 500 GB HDD, VGA Ge Force 740M 2 Gb. Perangkat lunak yang digunakan adalah sistem operasi Linux Ubuntu 16.04 LTS, *text editor* Visual Studio Code versi 1.13.1, DBMS MongoDB versi 3.2.10, bahasa pemrograman Javascript versi 1.7, *Platform* Node.js versi 6.7.0, *framework*: Express.js 4.14.1, dan *test API* Postman versi 5.0.2.

HASIL DAN PEMBAHASAN

Arsitektur Penelitian

Perancangan arsitektur perangkat lunak dibagi menjadi tiga bagian, yaitu data, *client*, dan *server* (Setiana 2016). Arsitektur perancangan tersebut kemudian dikembangkan dan digunakan selama penelitian. Hasil pengembangan arsitektur penelitian dapat dilihat pada Gambar 2. Bagian data menggunakan basis data non relasional. Bagian *server* menggunakan Google Maps API untuk menandai suatu posisi pengguna berada berdasarkan *latitude* dan *longitude*. *Server* Express.js digunakan untuk pengolahan data dengan format keluaran JSON. Bagian *client* dibagi menjadi dua, yaitu *mobile* dan web. Bagian web menggunakan *platform* AngularJS 2 untuk mengakses data dari *server* dan menampilkan informasi dari basis data, sedangkan pada *mobile* menggunakan *framework* Ionic 2 dengan *platform* AngularJS 2.



Gambar 2 Arsitektur Penelitian

Tahapan Penelitian

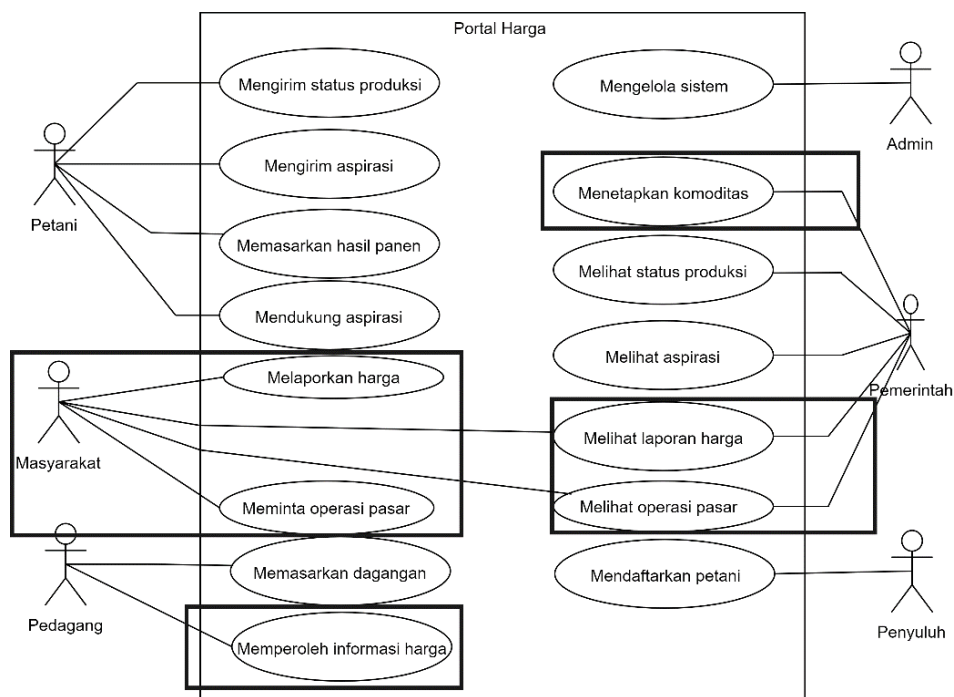
Proses pengembangan API dimulai pada bulan Februari 2017 dengan menggunakan arsitektur yang telah dikembangkan oleh tim. Metode yang digunakan, yaitu *extreme programming*. API yang dihasilkan dapat diakses oleh *front end* yang merupakan aplikasi berbasis web dan *mobile*. Tahapan yang metode *extreme programming* yang dilakukan oleh tim *back end* yaitu, *planning*, *design*, *coding*, dan *testing*. Keempat tahapan tersebut dilakukan iterasi sebanyak tiga kali.

Iterasi Pertama

Perencanaan (*Planning*)

Iterasi pertama melakukan *reengineering* modul dan fungsi–fungsi dasar yang dilakukan oleh Setyatama (2016). Modul yang dikembangkan penulis, yaitu modul komoditas, laporan harga, dan operasi pasar. Modul tersebut diakses oleh *stakeholder* pemerintah, masyarakat, dan pedagang. Fungsi dasar yang akan dibuat adalah *create*, *read*, *update*, dan *delete* untuk masing-masing modul.

User story dan *use case diagram* dibuat sesuai dengan modul dan *stakeholder* terkait. *User story* tersebut akan menjadi alur penggunaan sistem bagi pengguna. *Use case diagram* untuk mendefinisikan kebutuhan fungsional masing-masing *stakeholder* dapat dilihat pada Gambar 4. Pada *use case diagram* yang diberi kotak hitam merupakan fungsi tiap modul yang mengalami *reengineering* dan pengembangan. *Login* dilakukan oleh setiap *stakeholder* sebelum menjalankan fungsi–fungsi yang ada di dalam sistem. *Stakeholder* yang belum memiliki *account* dapat melakukan *register* untuk membuat *account*. *Use case diagram* untuk mendefinisikan kebutuhan fungsional masing-masing *stakeholder* dapat dilihat pada Gambar 3. Pada *use case diagram* yang diberi kotak hitam merupakan fungsi tiap modul yang mengalami *reengineering* dan pengembangan.



Gambar 3 Use case diagram iterasi pertama

Desain (Design)

Class diagram diimplementasikan pada basis data non relasional MongoDB. Teknik yang digunakan untuk mengubah *class* menjadi *collections* adalah *Object Document Mapper* (ODM). *Class* yang didefinisikan mewakili sebuah *document* yang otomatis akan tersimpan ke dalam sebuah *collections* (MongoDB 2017).

Pengodean (Coding)

Tahap *coding* melakukan pemisahan token dengan *string* Bearer. Token yang dikirim oleh *front end* berbasis web maupun *mobile* ketika mengakses API terdapat *string* Bearer. *String* Bearer muncul karena token yang dikirim untuk mengakses API berada pada *header* dengan *key authorization*.

Penelitian sebelumnya yang dilakukan oleh Setyatama (2016) menggunakan token yang masih bersifat statis, maka pada penelitian ini digunakan token JWT (JSON Web Token) yang bersifat dinamis. Token berfungsi untuk *security* data diletakkan di *header*. Token pertama kali diberikan ketika *user* melakukan *register* atau *login*. Token yang diberikan merupakan hasil *encode* dari data yang diperlukan untuk *request* API. *Front end mobile* memiliki status *login_type* = 1 yang berarti token tidak ada waktu kedaluwarsanya tetapi ketika *user* *logout* dari sistem maka token akan masuk ke dalam daftar *blacklist*. Nilai *front end website* status *login_type* = 0 yang berarti token memiliki waktu kedaluwarsa enam puluh menit, dan setiap kali melakukan *request* mendapatkan token baru. Token baru diberikan ketika *login_type* = 1, sedangkan untuk *login_type* = 0 nilai token baru yaitu “-”.

Pengujian (Testing)

Pengujian yang dilakukan, yaitu pengujian fungsi *create*, *read*, *update*, dan *delete* untuk setiap modul. *Unit test* dilakukan pada pengujian *role* pengguna. Berdasarkan hasil diskusi dengan tim, *role* setiap *stakeholder* sebagai berikut: *role* 1 adalah admin, *role* 2 adalah pemerintah, *role* 3 adalah penyuluh, *role* 4 adalah petani, *role* 5 adalah masyarakat, dan *role* 6 adalah pedagang. Hasil pengujian pada iterasi pertama dapat dilihat pada Tabel 2.

Tabel 2 Hasil pengujian iterasi pertama

<i>Class</i>	<i>Method</i>	Fungsi	URL	Status
Komoditas	POST	Menambah komoditas	komoditas/add	Berhasil
	GET	Melihat komoditas	komoditas/get	Berhasil
Komoditas	POST	Mengubah komoditas	komoditas/update	Berhasil
	POST	Menghapus komoditas	komoditas/delete	Berhasil
Laporan harga	POST	Menambah laporan harga	laporanHarga/add	Berhasil
	GET	Melihat laporan harga	laporanHarga/get	Berhasil
	POST	Mengubah laporan harga	laporanHarga /update	Berhasil
Operasi pasar	POST	Menghapus laporan harga	laporanHarga /delete	Berhasil
	POST	Menambah operasi pasar	operasiPasar/add	Berhasil
	GET	Melihat operasi pasar	operasiPasar/get	Berhasil
Operasi pasar	POST	Mengubah operasi pasar	operasiPasar /update	Berhasil
	POST	Menghapus operasi pasar	operasiPasar /delete	Berhasil

Iterasi Kedua

Perencanaan (*Planning*)

Modul operasi pasar dikembangkan sehingga pengguna dapat mendukung operasi pasar lain. Modul laporan harga dikembangkan sehingga dapat melihat laporan harga pada beberapa hari sebelumnya. Modul operasi pasar dan laporan harga dapat melihat *history* pengguna. Penelitian sebelumnya yang dilakukan oleh Mardika (2016) memiliki kekurangan, yaitu kendala apabila pengguna lupa *password*. Pengembangan modul *email* untuk mengatasi kekurangan tersebut.

Desain (*Design*)

Class operasi pasar memiliki empat fungsi yang dikembangkan, yaitu fungsi *vote*, *unvote*, *getPendukung*, dan *operasiku*. Fungsi *vote* dan *unvote* untuk mendukung dan membatalkan dukungan. Fungsi *getPendukung* untuk melihat siapa saja yang mendukung suatu operasi pasar. Fungsi *operasiku* untuk melihat *history* operasi pasar pengguna. *Class* laporan harga terdapat dua fungsi yang dikembangkan, yaitu fungsi *laporanku* dan *getDayLaporan*. Fungsi *laporanku* untuk melihat *history* laporan harga pengguna. Fungsi *getDayLaporan* untuk mendapatkan laporan harga beberapa hari sebelumnya. *Class email* dikembangkan satu fungsi yaitu *forgetPassword* yang memberikan *password* baru kepada pengguna yang dikirimkan melalui *email*.

Pengodean (*Coding*)

Iterasi kedua menambahkan field pendukung pada collections operasi pasar dengan tipe data array untuk menampung daftar pendukung. Ketika seorang pengguna mendukung suatu operasi pasar maka *user_id* pengguna tersebut akan dimasukkan ke dalam field array pendukung. Apabila seorang pengguna batal mendukung suatu operasi pasar maka *user_id* pengguna tersebut akan dikeluarkan dari field array pendukung. Implementasi modul *email* menggunakan *nodemailer* yang merupakan Gmail Google. Fungsi *forgetPassword* ketika diakses akan membuat *password* baru dan memperbarui *password* pengguna yang lama.

Password baru dikirimkan ke alamat *email* pengguna yang merupakan hasil *random string*. *Random string* yang dihasilkan sepanjang lima belas karakter.

Pengujian (Testing)

Fungsi yang dikembangkan pada modul operasi pasar, laporan harga, dan *email* dilakukan pengujian. Contoh pengujian, yaitu fungsi `forgetPassword` yang dilakukan dengan *login* menggunakan *password* baru yang dikirimkan melalui *email*. Hasil pengujian pada iterasi kedua dapat dilihat pada Tabel 3.

Tabel 3 Hasil pengujian iterasi kedua

<i>Class</i>	<i>Method</i>	Fungsi	URL	Status
Laporan harga	POST	Mendukung operasi pasar	operasiPasar/pendukung/add	Berhasil
	POST	Membatalkan dukung operasi pasar	operasiPasar/pendukung/delete	Berhasil
	GET	Melihat pendukung operasi pasar	operasiPasar/pendukung/get/:operasiPasar_id	Berhasil
	GET	Melihat <i>history</i> laporan harga	operasiPasar/operasi/get/:user_id	Berhasil
Operasi pasar	GET	Melihat <i>history</i> operasi pasar	laporanHarga/get/laporan/:user_id	Berhasil
	GET	Melihat laporan harga beberapa hari sebelumnya	laporanHarga/get/day/:day	Berhasil
Mail	POST	Mengirim password	email/forgetPassword	Berhasil

Iterasi Ketiga

Perencanaan (Planning)

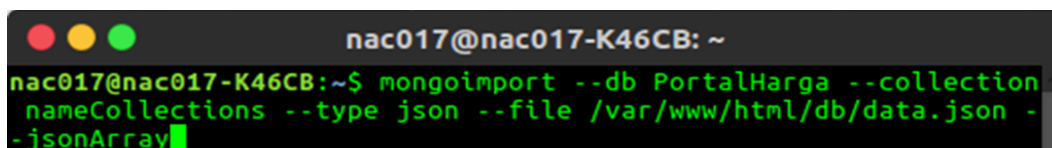
Iterasi ketiga mengembangkan laporan harga, operasi pasar, *email*, dan merancang modul lokasi. Modul laporan harga dapat melihat laporan harga selama satu minggu ataupun selama satu bulan. Modul operasi pasar pengguna dapat memberikan tanggapan. Modul lokasi bertujuan untuk memberikan pilihan lokasi kepada pengguna pada saat *register* atau saat memberikan informasi laporan harga atau operasi pasar sesuai dengan lokasi pengguna berada.

Desain (Design)

Modul lokasi terdiri atas *class* jenis, provinsi, kabupaten, kecamatan, dan kelurahan. *Class* kelurahan memiliki ketergantungan terhadap *class* kecamatan, *class* kecamatan memiliki ketergantungan terhadap *class* kabupaten, dan *class* kabupaten memiliki ketergantungan terhadap *class* provinsi. Modul laporan harga terdapat dua fungsi yang dikembangkan, yaitu fungsi `getWeek` dan `getMont`. Modul operasi pasar terdapat tiga fungsi yang dikembangkan, yaitu `addTanggapan` untuk memberikan tanggapan, `delTanggapan` untuk menghapus tanggapan, dan `getTanggapan` melihat siapa saja yang memberikan tanggapan. Modul *email* terdapat empat fungsi yang dikembangkan, yaitu `getMailVerify`, `postVerify`, `postReSendVerify`, dan `postReNewPassword`. Fungsi `getVerify` untuk mengirim *email* verifikasi.

Pengkodean (*Coding*)

Iterasi ketiga membuat API untuk modul lokasi menggunakan data lokasi seluruh Indonesia dalam format SQL. Data format SQL kemudian diubah menjadi format JSON untuk masing-masing tabel menggunakan phpMyAdmin. Data dalam format JSON kemudian dimasukan ke dalam basis data MongoDB dengan menggunakan *terminal* untuk masing-masing tabel menjadi *collections* yang bersesuaian. *Query* yang digunakan untuk memasukan data ke dalam *collections* MongoDB adalah *mongoimport*. Contoh *query* memasukan data provinsi ke dalam basis data MongoDB yaitu dapat dilihat pada Gambar 8. *Query* tersebut memasukan data provinsi dalam format JSON yang berada dalam folder `var/www/html/db` ke dalam *collections* provinsi di basis data PortalHarga.



```
nac017@nac017-K46CB: ~
nac017@nac017-K46CB:~$ mongoimport --db PortalHarga --collection
nameCollections --type json --file /var/www/html/db/data.json -
-jsonArray
```

Gambar 8 *Code* memasukkan data format JSON ke dalam basis data MongoDB

Pengujian (*Testing*)

Pengujian pada modul lokasi untuk mendapatkan provinsi seluruh Indonesia, kabupaten atau kota berdasarkan provinsi yang dipilih, kecamatan berdasarkan kabupaten atau kota yang dipilih, dan kelurahan berdasarkan kecamatan yang dipilih. Modul *email* dilakukan pengujian dengan memverifikasi account melalui link yang dikirim ke pengguna melalui *email*. Detil pengujian iterasi ketiga dapat dilihat pada Tabel 4.

Tabel 4 Hasil pengujian iterasi kedua

<i>Class</i>	<i>Method</i>	Fungsi	URL	Status
Provinsi	GET	Melihat provinsi	lokasi/provinsi	Berhasil
Kabupaten	GET	Melihat kabupaten	lokasi/kabupaten/:id_prov	Berhasil
Kecamatan	GET	Melihat kecamatan	lokasi/kecamatan/:id_kab	Berhasil
Kelurahan	GET	Melihat kelurahan	lokasi/kelurahan/:id_kec	Berhasil
Mail	GET	Mengirim email verifikasi	/user/email/getMailVerify	Berhasil
	POST	Mengirim ulang email verifikasi	/user/email/validate/resend	Berhasil
	POST	Mengirim <i>link</i> ke halaman <i>forget password</i>	/user/email/forgetPassword	Berhasil
	POST	Memperbarui <i>password</i>	/user/email/reNewPassword	Berhasil
Laporan harga	GET	Melihat laporan harga minggu ini	laporanHarga/getWeek	Berhasil
	GET	Melihat laporan harga bulan ini	laporanHarga/getMonth	Berhasil

Tabel 4 Lanjutan

<i>Class</i>	<i>Method</i>	Fungsi	URL	Status
Operasi pasar	POST	Memberi tanggapan operasi pasar	operasiPasar/tanggapan/add	Berhasil
	POST	Menghapus tanggapan operasi pasar	operasiPasar/tanggapan/detele	Berhasil
	GET	Melihat siapa saja yang memberi tanggapan	operasiPasar/get/:operasiPasar	Berhasil

SIMPULAN DAN SARAN

Simpulan

Penelitian ini berhasil mengembangkan REST API menggunakan Node.js dan basis data non relasional dengan menggunakan metode Extreme Programming dan dilakukan iterasi sebanyak tiga kali. Pengembangan API mampu menghubungkan enam aktor, yaitu admin, pemerintah, penyuluh, petani, masyarakat, dan pedagang. API yang dihasilkan sudah terintegrasi dengan *email* dan lokasi seluruh Indonesia. API sudah dapat diakses oleh front end berbasis mobile maupun web.

Saran

Penelitian selanjutnya pada tahap pengodean perlu dianalisis efisiensi algoritme agar *execution times* lebih cepat. Penambahan modul *region* agar pemerintah bisa memberikan patokan harga yang sesuai berdasarkan wilayahnya, melihat laporan harga sesuai dengan lokasi pengguna berada. Selain itu, penelitian selanjutnya dapat mengimplementasikan fitur JOIN yang ada pada basis data non relasional.

DAFTAR PUSTAKA

- [APJII] Asosiasi Penyelenggara Jasa Internet Indonesia. 2014. Profil pengguna internet Indonesia 2014 [Internet]. [diunduh 2016 Des 22]. Tersedia pada: <http://www.slideshare.net/internetsehat/profil-pengguna-internet-indonesia-2014-riset-oleh-apjii-dan-puskakom-ui>.
- [JSON] JavaScript Object Notation. 2002. Introduction JSON [internet]. [diunduh 2017 Jul 27]. Tersedia pada: <http://json.org/json-id.html>.
- [Kementan] Kementerian Pertanian. 2015. Renstra kementan tahun 2015-2019 [Internet]. [diunduh 2017 Mei 13]. Tersedia pada: http://www.pertanian.go.id/file/RENSTRA_2015-2019.pdf.
- Mardika SB. 2016. Pengembangan REST API modul produsen komoditas pertanian pada aplikasi portal harga [skripsi]. Bogor (ID): Institut Pertanian Bogor.
- [MongoDB]. 2017. MongoDB Documentation Release 3.0.7. [internet] [diunduh 2017 Aug 7] Tersedia pada <https://docs.mongodb.com/v3.4/mongodb-manual-v3.4.epub>.
- Pautasso C. 2008. REST vs SOAP Making the Right Architectural Decision. Amsterdam (NL): SOA Symposium.
- Pressman RS. 2010. *Software Engineering : A Practitioner's Approach Ed ke-7*. Boston (US): Mc Graw Hill.
- Setyatama D. 2016. Pengembangan aplikasi pelaporan harga komoditas pertanian berbasis *mobile* menggunakan REST API [skripsi]. Bogor (ID): Institut Pertanian Bogor.