**Proposal Tugas Akhir**

**Pengembangan Modul Pelaporan Harga Komoditas Pertanian untuk Masyarakat dan Pedagang Menggunakan REST API**

NUGROHO AGUNG CAHYONO (G64130039)\*, DEAN APRIANA RAMADHAN

**ABSTRAK**

Sektor pertanian di Indonesia menjadi sumber mata pencaharian maupun sebagai penopang pembangunan. Kesenjangan informasi yang terjadi di antara pemerintah, petani, pedagang, dan masyarakat menyebabkan fluktuasi harga. Perkembangan internet dapat dijadikan sebagai peluang untuk mengatasi kasus fluktuasi tersebut dengan menyediakan media yang mampu menghubungkan petani, masyarakat, pedagang, dan pemerintah. Penelitian ini bertujuan untuk memberikan informasi dan media komunikasi antara pemerintah, petani, masyarakat, dan pedagang sehingga stabilitas harga dapat tercapai. Penelitian ini melanjutkan penelitian sebelumnya yaitu pengembangan modul pelaporan harga komoditas pertanian pada sisi pengguna admin dan pemerintah menggunakan REST API, basisdata yang digunakan yaitu MySQL. Komoditas pertanian yang digunakan pada penelitian tersebut terdiri dari beras, daging sapi, cabai, kedelai, dan bawang merah. Basis data yang akan digunakan dalam penelitian ini yaitu basisdata non relasional. Metode yang digunakan yaitu metode *Extreme Programming* yang merupakan bagian dari *Agile Software Engineering.*

Kata kunci: basis data non relasional*,* *extreme programming*,REST API.

***ABSTRACT***

Agricultural sector in Indonesia become a source of livelihood as well as the support of development. Information gap that occurs between government, farmers, traders and the public cause price fluctuations. The development of the Internet can be used as an opportunity to address the case of these fluctuations by providing a media that can connect farmers, communities, merchants, and government. This study aims to provide information and media communication between government, farmers, communities, and merchants that price stability can be achieved. This study continues the previous research reporting module development in agricultural commodity prices on the government side and the admin user using the REST API, the database used is MySQL. Agricultural commodities used in the study consisted of rice, beef, chili, beans and onions. The database that will be used in this study is a non-relational database. The method used is the method of Extreme Programming that is part of the Agile Software Engineering

Keywords: : *extreme programming*, *non-relational database*, REST API.

Departemen Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan alam, Institut pertanian Bogor, Bogor 1660

\*Mahasiswa Program Studi Ilmu Komputer, FMIPA-IPB

**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 petanian 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.

Kesenjangan komunikasi yang terjadi antara petani dengan pemerintah merupakan permasalahan yang terjadi dalam mengelola dan mengembangkan sektor pertanian. Kesenjangan tersebut dimanfaatkan oleh pihak ketiga, yaitu tengkulak karena mereka membeli hasil panen dari petani dengan harga yang rendah dan dijual kembali dengan harga yang tinggi. Kondisi tersebut yang membuat terjadinya fluktuasi harga.

Menurut Undang-undang Republik Indonesia Nomor 8 Tahun 1999 tentang perlindungan konsumen menyatakan bahwa konsumen berhak untuk memilih barang atau jasa serta mendapatkan barang atau jasa tersebut sesuai dengan nilai tukar dan kondisi serta jaminan yang dijanjikan. Masyarakat berhak mendapatkan perlindungan jika harga komoditas pertanian tidak stabil. Harga komoditas pertanian yang selalu mengalami perubahan membuat petani merasa kesulitan untuk memasarkan hasil pertaniannya. Kondisi tersebut disebabkan jarak sumber informasi yang jauh, sehingga informasi yang diterima oleh petani tidak *up-to-date* (Saputra 2015)*.*

Harga komoditas pertanian yang tidak menentu membuat pedagang kesulitan untuk menentukan harga jual yang sesuai. Harga yang tidak menentu disebabkan oleh besarnya jumlah penawaran dan permintaan. Jumlah penawaran yang tinggi atau rendah disebabkan oleh waktu terjadinya musim panen. Faktor cuaca dan serangan hama menjadi faktor yang membuat tingginya gagal panen (Wihono 2009).

Pesatnya perkembangan teknologi informasi dapat digunakan untuk mengatasi permasalah yang terjadi di sektor pertanian. Teknologi informasi yang dapat dikembangkan salah satunya, yaitu media internet atau komunikasi dunia maya (Elian 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 2014)*.*

Penelitian ini melanjutkan penelitian sebelumnya, yaitu pengembangan modul pelaporan harga komoditas pertanian pada sisi pengguna admin dan pemerintah menggunakan REST API (Setiana 2016). Pengembangan token OAuth menjadi token yang dinamis untuk keamanan data diperlukan pada sistem tersebut (Setyatama 2016). Penelitian tentang sistem informasi pelaporan harga sebelumnya telah dilakukan, di antaranya oleh Saputra (2015) dengan membangung aplikasi *Short Message Service* (SMS) *center* untuk informasi harga komoditi hasil pertanian Kabupaten Ogan Ilir. Sms yang masuk maupun keluar dikelola oleh admin, kemudian diintegrasikan dengan dinas terkait sesuai dengan harga komoditi pertanian (Saputra 2015). Penelitian yang dilakukan oleh Ahmad (2015) tentang perancangan aplikasi komoditas pertanian berbasis android. Pengguna dapat mengakses sistem untuk mendapatkan informasi komoditas, informasi harga, informasi cuaca, dan berita.

Penelitian ini mengembangkan modul pelaporan harga komoditas pertanian untuk masyarakat dan pedagang menggunakan REST API. Modul yang dikembangkan mampu menjadi media informasi antara *stakeholder* terkait dan mampu menyediakan informasi seputar harga komoditas pertanian kepada masyarakat secara *up-to-date.* Pengembangan modul ini menggunakan REST API dengan memanfaatkan *framework* Express.js dan basis data non relasional dengan keluaran yang dihasilkan berupa format JSONyang fleksibel dan dapat diakses oleh berbagai *platform,* baik web maupun *mobile*.

**Perumusan Masalah**

Bagaimana cara mengembangkan modul pelaporan harga komoditas pertanian, sehingga mampu mengolah informasi dan menyampaikannya kepada *stakeholder* yang bersangkutan.

**Tujuan Penelitian**

Mengembangkan modul pelaporan harga komoditas pertanian menggunakan REST API dan *database non-relational,* dan menghubungkan antar *stakeholder* terkait sehingga semua *stakeholder* saling terintegrasi.

**Manfaat Penelitian**

Manfaat dari penelitian ini adalah:

1. Dari sisi pedagang:

* 1. Informasi harga komoditas pertanian bisa diketahui secara *real-time*.
  2. Mampu menentukan harga jual komoditas pertanian.

1. 2. Dari sisi masyarakat:
   1. Masyarakat mengetahui harga yang beredar di pasar sehingga mampu menyesuaikan dengan kebutuhannya.
   2. Masyarakat ikut berperan dalam pengendalian harga komoditas pertanian dengan melaporkan harga di pasar secara langsung.
2. 3. Dari sisi penyuluh

a Mampu menjembatani program dari pemerintah sehingga sampai ke petani.

b Mengetahui kegiatan yang dilakukan dan permasalahan yang dihadapi oleh petani.

**Ruang Lingkup Penelitian**

Ruang lingkup dari penelitian ini adalah:

1. Sistem tidak melakukan validasi terhadap masukan yang diberikan oleh pengguna.
2. Komoditas pertanian yang digunakan sama dengan penelitian sebelumnya yaitu terdiri dari beras, daging sapi, cabai, kedelai, dan bawang merah.
3. Mengembangkan modul untuk masyarakat, pedagang dan penyuluh.

**METODE**

**Data Penelitian**

Data yang digunakan pada penelitian ini sama dengan penelitian pengembangan modul pelaporan harga komoditas pertanian pada sisi pengguna admin dan pemerintah menggunakan REST API. Data yang diperoleh tidak dilakukan verifikasi data. Data yang digunakan merupakan hasil observasi dari internet dan Badan Pusat Statistik, kemudian dilakukan analisis MVP (*Minimum Variable Priduct*).

**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 (Rohman 2015). *Extreme Programming* (XP) merupakan metodologi pengembangan perangkat lunak yang tanggap terhadap perubahan kebutuhan pengguna, sehingga meningkatkan kualitas perangkat lunak (Pressman 2010). Perubahan *requirements* dari pengguna dapat segera ditanggapi oleh pengembang meskipun pengembangan perangkat lunak sudah dilakukan (Oktaviani dan Hutrianto 2016). Tahapan dalam *Extreme Programming*,yaitu *planning, design, coding,* dan *testing* dapat dilihat pada Gambar 1*.*

C:\Users\fiqih\AppData\Local\Microsoft\Windows\INetCacheContent.Word\xp.png

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

* + 1. **Perencanaan (*Planning*)**

Perencanaan *requirement* yang diperlukan yang akan dibuat menjadi modul-modul dan fungsi-fungsi pada sistem. Pembuatan *user story* dan *use case diagram* sesuai *stakeholder* yang terkait. *User story* merupakan deskripsi pendek dan sederhana yang disampaikan berdasarkan sudut pandang pihak yang menginginkan kapabilitas baru mengenai sebuah fitur pada sebuah sistem (Cohn 2014). *Use case* merupakan deskripsi fungsi dari sebuah sistem berdasarkan perspektif pengguna (Basuki 2011). Iterasi yang akan dilakukan selama penelitian sebanyak tiga kali.

* 1. **Desain (*Design*)**

Perancangan yang dilakukan mencakup pembuatan *class, responsibilities,* dan *collaboration* (CRC) *cards* dan *sequence diagram.* Perancangan dilakukan secara berkelanjutan selama proses pengembangan sesuai dengan rencana iterasi. *Class* diagramadalah gambaran struktur sistem berdasarkan dari pendefinisian *class-class* yang akan dibuat dalam membangun sistem (Setiady 2013). *Sequence* diagramyang dibuat pada tahap desain merupakan gambaran interaksi dan komunikasi diantara objek-objek disusun berdasarkan urutan waktu (Haviludding 2011). *Spike solution prototype* dilakukan selama tahap desain untuk meminimalkan risiko ketika memulai pengembangan (Pressman 2010).

* 1. **Pengkodean (*Coding*)**

Selama proses pengembangan perangkat lunak anggota tim bekerja secara berpasangan dan saling berbagi tugas. Tahap pengkodean dilakukan *continuous integration,* yaitu apabila terjadi perubahan *requirement* selama proses pengembangan, maka harus segera diintegrasikan (Oktaviani dan Hutrianto 2016). Ketika sebuah tugas telah selesai dikerjakan maka akan diintegrasikan dengan bagian lainnya pada keseluruhan sistem. *Pair programming* dilakukan dengan bagian *back end* yang dilakukan oleh rekan satu tim, yaitu Fiqih Nur Ramadhan.

* 1. **Pengujian (*Testing*)**

Penyusunan *acceptance* *testing* yang berdasarkan fungsi atau fitur yang diperoleh dari *user story* pada proses perencanaan (Pressman 2010). Pengujian dilakukan secara *black-box* menggunakan *Postman.* Pengujian dengan perintah HTTP menggunakan fungsi POST dan GET*.* Pengujian dilakuakan untuk mengetahui apakah API berjalan baik atau tidak.Keluaran data yang diperoleh setelah mengakses API dalam format JSON.

**Lingkungan Pengembangan**

Spesifikasi yang digunakan pada penelitian ini, perangkat keras dan perangkat lunak. Perangkat keras pengembangan, yaitu *processor* Intel® Core™ i3-3217u, RAM 4 GB, 500 GB HDD, dan VGA Ge Force 740M 2 Gb. Perangkat lunak pengembangan, yaitu sistem operasi Linux Ubuntu 16.04 LTS, t*ext editor* Visual Studio Code, DBMS MongoDB, bahasa pemrograman javascript, *platform* Node.js, *framework* Express.js, *test* API Postman.

**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 daribasis 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 dan menggunakan metode *Extreme Programming*. Sistem yang dikembangkan menghasilkan API yang akan diakses oleh *front end* yang merupakan aplikasi berbasis web dan *mobile*. Berdasarkan penyataan Pressman(2010), metode *exterme programming* terdiri dari lima tahapan namun yang dilakukan oleh tim *back end* hanya empat tahapan yaitu, *planning*, *design*, *coding*, dan *testing*. Keempat tahapan tersebut dilakukan iterasi sebanyak tiga kali.

**1 Iterasi Pertama**

* 1. **Perencanaan (*Planning*)**

Perencanaan awal dilakukan oleh tim untuk menentukan *stakeholder* dan *role* dari setiap *stakeholder*. Berdasarkan hasil diskusi dengan tim, terdapat enam *stakeholder*, yaitu *admin*, pemerintah, penyuluh, petani, masyarakat, dan pedagang. *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*.*

*User story* dibuat dengan tujuan untuk mempermudah dalam proses pengembangan sistem. Salah satu *user story*, yaitu *user story* masyarakat. Detail *user story* masyarakat dapat dilihat pada tabel 2. *User story* tersebut akan menjadi alur penggunaan sistem bagi pengguna.

Tabel 2 *User story* masyarakat

|  |  |  |
| --- | --- | --- |
| *Actor* | *Task* | *User story* |
| Masyarakat | Memberikan laporan harga | Pengguna memberikan laporan harga ketika berada pada suatu lokasi pasar. Penguna memasukkan harga suatu komoditas, sehingga bisa diketahui oleh pengguna lain. |
|  | Melihat laporan harga | Masyarakat memperoleh informasi harga komoditas beserta lokasinya. Informasi tersebut berasal dari masukan pengguna lain yang sudah melaporkan informasi harga pada suatu lokasi, sehingga masyarakat bisa mendapatkan harga yang sesuai dengan kebutuhan. |
|  | Meminta operasi pasar | Harga komoditas yang tinggi pada suatau lokasi, maka masyarakat dapat memberi pesan kepada pemerintah untuk melakukan operasi pasar untuk suatu komoditas pada suatu lokasi tersebut. |
|  | Melihat operasi pasar | Masyarakat dapat melihat operasi pasar yang telah diminta oleh pengguna lain, sehingga pengguna tidak akan berbelanja suatu komoditas pada lokasi tersebut karena harga yang tinggi. |

Pembuatan *use case* digrambertujuan untuk mendefinisikan kebutuhan fungsional masing masing *stakeholder*. Setiap *stakeholder* perlu *login* sebelum menjalankan fungsi-fungsi yang ada di dalam sistem. *Stakeholder* yang belum memiliki *account* dapat melakukan *register* untuk membuat *account*.

Pembuatan modul dan fungsi-fungsi dasar berdasarkan kebutuhan fugsional *stakeholder* yang sudah ditentukan*.* Modul yang sudah ditentukan dikerjakan dengan cara dibagi dengan tim *back end* sebelum nantinya diintegrasikan menjadi satu. Penulis mengerjakan modul operasi pasar, laporan harga, dan komoditas. Fungsi dasar yang akan dibuat adalah CRUD.

* 1. **Desain (Design)**

Perancangan dan pembuatan *class* diagram dilakukan dalam tahap desain*.* Pembuatan *class* diagram bertujuan untuk memberikan gambaran tentang sistem dan mempermudah dalam pengembangan sistem. *Class diagram* diimplementasikan pada basis data nonrelasional mongoDB dengan setiap *class* yang dibuat akan disimpan dalam sebuah *collections* di dalam *basis data*.

Penentuan atribut pada sebuah *class* menggunakan konsep *Big-O*, hal ini bertujuan untuk mengatasi segala kemungkinan atribut yang dimiliki oleh *object* termasuk turunan *class*nya. *Class user* atributnya menggunakan konsep *Big-O* untuk menampung segala kemungkinan atribut yang dimiliki oleh setiap jenis *user. Class* admin, pemerintah, penyuluh, petani, masyarakat, dan pedagang merupakan turunan dari *class* *user.*

Tahap desain juga dilakukan perencanaan dan pembuatan *sequence* diagram.Salah satu *sequence* diagram merupakan fungsi tambah laporan pada modul laporan harga. Alur pekerjaan yang akan dilakukan adalah menambah laporan harga, masyarakat memasukkan data harga suatu komoditas, lokasi, dan token ke API, token yang dikirim akan dicek oleh API. Berdasarkan token API juga mengecek *role* pengguna, jika token dan *role* sesuai maka API akan menyimpan data ke dalam basis data dan basis dataakan mengecek apakah ada *error,* API menerima status penyimpanan apakah berhasil atau *error* dari basis data dan mengirimkannya ke masyarakat.

* 1. **Pengkodean (*Coding*)**

Selama tahap *coding* melakukan *pair programming* dengan rekan satu tim *back end*, yaitu Fiqih Nur Ramadhan. *Pair programming* bertujuan untuk mendukung *continuous integration* yang dilakukan ketika terjadi perubahan *requirement* selama proses pengembangan agar cepat terselesaikan kemudian diintegrasikan dengan keseluruhan bagian pada sistem. GitHub digunakan untuk mempermudah *pair programming* dan penggabungan *code* pada tim *back end*. Reposirori GituHub yang digunakan, yaitu https://github.com/ryanbaskara/backendPH. Ketika sebuah tugas telah selesai dikerjakan maka melakukan *push* ke repositori GitHub, kemudian anggota tim yang lain dan *server* akan melakukan *pull* terhadap repositori GitHub tersebut, sehingga semua saling terintegrasi.

*Method* yang digunakan untuk mengakses REST API, yaitu POST dan GET. *Port* yang digunakan untuk mengkases REST API adalah 5000. Penggunaan *method* POST dan GET diatur di dalam *route*. *Method* POST digunakan untuk memasukkan data ke dalam *basis data*, sedangkan *method* GET untuk memperoleh data dari basis data. *Method* POST data yang akan dimasukkan diletakkan di *body.* Contoh REST API dengan menggunakan *method* POST, yaitu dengan mengakses URL <https://ph.yippytech.com:5000/laporanHarga/add>. Contoh REST API dengan *method* GET, yaitu dengan mengakses URL <https://ph.yippytech.com:5000/laporanHarga/get>. Format keluaran setelah mengkases API dengan *method* POST atau GET terdiri dari JSON *array* yang diawali dan diakhiri dengan kurung siku (“[]”)dan JSON *object* yang yang diawali dan diakhiri dengan kurung kurawal (“{}”).

Token yang digunakan dalam sistem adalah JWT (JSON Web Token) yang bersifat dinamis. Token berfungsi untuk *security* data diletakkan di *header* dengan *key* token. 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 kadaluarsanya tetapi ketika *user logout* dari sistem maka token akan masuk ke dalam daftar *blacklist*. *Front end* *website* status *login\_type* = 0 yang berarti token memiliki waktu kadaluwarsa 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 yaitu “-”.

* 1. **Pengujian (*Testing*)**

Tahap setelah *coding* adalah tahap *testing* untuk menguji setiap fungsi dari serangkaian modul yang telah dibuat. Pengujian menggunakan metode *black-box* dan dilakukan secara internal kemudian diintegrasikan dengan anggota tim *front end*,yaitu Ryan Baskayara dan Irfan Rafii. Modul yang akan dilakukan *testing*,yaitu operasi pasar, laporan harga, dan komoditas.

Pengujian pertama adalah pengujian token, mengecek ketersediaan token dan memverifikasi token yang diberikan. Pengujian selanjutnya, yaitu mengecek dan memverifikas *role* yang berasal dari hasil *decode* token. *Role* digunakan untuk mengakses suatu fungsi tertentu. Pengujian terakhir berhubungan dengan basis data, yaitu mengambil data dari basis data, jika data yang diminta tidak tersedia pada basis data atau kosong, maka gagal menyimpan data ke dalam basis data*,* dan parameter yang diberikan tidak sesuai yang ada di basis data.

**2 Iterasi Kedua**

1. **Perencanaan (*Planning*)**

Tahap perencanaan menentukan format keluaran JSON yang konsisten berdasarkan *request* dari *front end*. Berdasarkan hasil diskusi dengan tim, format keluaran JSON terdiri dari status, data, *message*, dan token. Status merupakan respon yang diberikan oleh server setelah dilakukan *request*. Status berisi nilai *integer* yang merupakan respon yang diberikan oleh *server* terhadap *request* yang dilakukan, 200 berarti suskes, 204 berarti data kosong, 401 berarti *unauthorized* yang disebabkan oleh token yang tidak valid, dan 408 berarti *request timeout.* Data berisi *array* yang terdiri atas *object-object* mapun *array* hasil *query* dari *basis data* sesuai dengan *requirement* yang dibutuhkan oleh *front end.* *Message* merupakan *string* yang berisi informasi setelah melakukan *request* ke *server*. Pada iterasi kedua, token tetap diletakkan di *header*, namun *key* yang digunakan adalah *Authorization.*

Iterasi kedua pada modul operasi pasar terdapat empat fungsi baru. Fungsi untuk mendukung dan batal dukung operasi pasar yang telah di unggah oleh pengguna lain. Fungsi untuk melihat siapa yang telah mendukung suatu operasi pasar. Fungsi untuk melihat *history* operasi pasar yang telah diunggah.

Modul laporan harga pada iterasi kedua terdapat dua fungsi baru. Fungsi untuk melihat *history* operasi pasar yang telah diunggah. Fungsi untuk melihat laporan harga yang masuk ke sistem pada beberapa hari sebelumnya.

Modul email diperlukan untuk mengintegrasikan sistem dengan email agar sistem lebih dinamis. Fungsi yang ada pada modul email, yaitu *forget password*. Fungsi tersebut akan dijalankan ketika pengguna lupa *password*.

1. **Desain (*Design*)**

Perancangan *class* diagram dan *sequence* diagramdisesuaikan dengan modul dan fungsi yang ditambahkan pada tahap perencanaan. *Class* operasi pasar terdapat empat fungsi baru, yaitu fungsi *vote*, *unvote*, getPendukung, dan operasiku. *Class* laporan harga terdapat dua fungsi baru, yaitu fungsi laporanku dan getDayLaporan. *Class* email hanya terdapat satu fungsi yaitu *forgetPassword*.

Salah satu *sequence* diagram adalah fungsi *forgetPassword*. Pengguna memasukkan *username* kemudian API akan membuat *password* baru untuk pengguna. API akan menimpa *password* lama pengguna yang ada di basis data. API juga akan mengirimkan *password* baru kepada pengguna melalui email.

1. **Pengkodean (*Coding*)**

Tahap *coding* pada iterasi kedua dilakukan pemisahan token dengan *string* Bearer. Token yang dikirim oleh *mobile* ketika mengkases API terdapat *string* Bearer. *String* Bearer dikrim oleh *mobile* karena token yang dikirim untuk mengakses API berada pada *header* dengan *key Authorization.*

Iterasi kedua pada *collections* operasi pasar ditambahkan *field* pendukung 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.

Iterasi kedua pada modul laporan harga untuk fungsi getDayLaporan dilakukan *time out* selama 400 *miliseconds*. *Time out* selama 300 *miliseconds* untuk mendapatkan id\_laporan pada beberapa hari sebelumnya kemudian dimasukkan ke dalam *array*. Sisa *time out* 100 *miliseconds* untuk mendapatkan data laporan harga sesuai id\_laporan yang sudah disimpan di *array*.

Implementasi modul email menggunakan *npm* *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.

1. **Pengujian (*Testing*)**

Pengujian yang dilakukan pada iterasi kedua sama dengan pengujian pada iterasi pertama, menggunakan pengujian *black box*. Pengujian pada iterasi kedua dilakukan oleh tim secara internal untuk semua fungsi pada tiap modul. Setelah pengujian, kemudian diintegrasikan dengan anggota tim *front end* dansemua fungsi yang telah diuji dapat berjalan dengan baik.

**3 Iterasi Ketiga**

1. **Perencanaan (*Planning*)**

Berdasarkan analisis *requirement* yang dilakukan oleh tim, pada iterasi ketiga diperlukan modul lokasi dan mengembangkan modul email. Modul lokasi bertujuan untuk memberikan pilihan lokasi kepada pengguna pada saat *register*. Modul email dikembangkan untuk verifikasi *account*. Verifikasi *account* dikirim melalui email dan jika tidak diverifikasi maka tidak bisa *login*.

Pengguna setelah selesai mendaftar akan mendapatkan email yang berisi *link* untuk memverifikasi *account*. Pengguna yang belum melakukan verifikasi *account* tidak bisa *login* ke dalam sistem. *Link* yang dikirim memiliki waktu *expired*. Pengguna dapat meminta untuk mengirim lagi email verifikasi.

1. **Desain (*Design*)**

Tahap desain membuat *class* baru, yaitu *class* lokasi dan mengembangkan *class* email. *Class* lokasi terdapat fungsi yang dapat digunakan oleh pengguna untuk memilih provinsi, kabupaten, kecamatan, dan kelurahan. *Class* email terdapat empat fungsi, pertama *forgetPassword* berubah menjadi mengirimkan email ke pengguna jika pengguna lupa dengan *password* dengan memberikan suatu *link* yang disisipi parameter hasil *encode* *username* ke suatu halaman web. Kedua funsgi *reNewPassword* yang merupakan lanjutan dari fungsi *forgetPassword* untuk memperbarui *password* pengguna berdasarkan *username* yang sesuai. Email yang dikirimkan kepada pengguna berisi link yangakan mengakses suatu halaman dan pengguna diminta untuk memasukkan *password* baru di kolom yang sudah disediakan. Fungsi ketiga, getMailVerify merupakan fungsi yang akan mengirim email kepada pengguna ketika pengguna pertama kali mendaftar atau ketika *login*. Keempat fungsi untuk memverifikasi *account* melalui *link* yang dikirim ke pengguna melalui email.

1. **Pengkodean (*Coding*)**

Iterasi ketiga membuat API untuk modul lokasi menggunakan data lokasi seluruh Indonesia dalam format sql. Data format sql kemudia diubah menjadi format JSON untuk masing-masing tabel menggunakan Phpmyadmin. Data format sql terdiri dari lima tabel yaitu, jenis, provinsi, kabupaten, kecamatan, dan kelurahan. Data dalam format JSON kemudian dimasukkan ke dalam basis data mongoDB dengan menggunakan *terminal* untuk masing-masing tabel menjadi *collections* yang bersesuaian.

*Pair programming* dilakukan antara fungsi *getMailVerify* dengan fungsi *register*. Email yang dikirm ke pengguna berisi *link* yang diintegrasikan dengan fungsi *postVerify*. Fungsi *postMailVerify* untuk mengubah *field isValidate* pada *collections users* agar bernilai *true* yang secara *default* bernilai *false.*

1. **Pengujian (*Testing*)**

Pengujian yang dilakukan dengan menguji setiap fungsi yang telah dibuat pada modul lokasi dan modul email menggunakan metode *black box.* Pengujian dilakukan secara internal oleh tim dan diintegrasikan dengan *requirement* *front end*. Pengujian pada modul lokasi menguji fungsi untuk mendapatkan provinsi seluruh Indonesia, kabupaten atau kota berdasarkan provinsi yang dipilih, kecamatan berdasarkan kabupaten atau kota yang dilih, dan kelurahan berdasarkan kecamatan yang dipilih. Pengujian pada modul email menguji fungsi *forget password*, memperbarui password, mengirim email verifikasi *account*, dan memverifikasi *account*.

**SIMPULAN DAN SARAN**

**Simpulan**

Penelitian ini berhasil mengembangkan REST API menggunakan Node.js dan basis data nonrelasional 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 Indoensia. API dapat diakses di ph.yippytech.com:5000 dan sudah diintegrasikan dengan *front end* berbasis *mobile* maupun web*.*

**Saran**

Penelitian selanjutnya pada tahap pengkodean perlu dianalisis efisiensi algoritma agar *execution times* lebih cepat. Penambahan modul *region* agar pemerintah bisa memberikan patokan harga yang sesuai berdasarkan wilayahnya. Selain itu, memaksimalkan fitur-fitur yang ada pada basis data nonrelasional.

**DAFTAR PUSTAKA**

Andarwati SR, Sankarto BS. 2005. Pemenuhan kepuasan penggunaan internet oleh peneliti Badan Litbang Pertanian di Bogor. *Jurnal Perpustakaan Pertanian*. 14(1): 10-13.

Ashana LN. 2015. Pertukaran data antara Rstudio dan MongoDB pada sistem informasi geografis untuk kasus pertanian Indonesia [skripsi]. Bogor (ID): Institut Pertanian Bogor.

[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>.

Cohn M. 2014. User Stories [Internet]. [diunduh 2017 Jan 16]. Tersedia pada: <https://www.mountaingoatsoftware.com/agile/user-stories>.

Elian N, Lubis DP, Rangkuti PA. 2014. Penggunaan internet dan pemanfaatan informasi pertanian oleh penyuluh pertanian di Kabupaten Bogor wilayah Barat. *Jurnal Komunikasi Pembangunan.* 12(2):105-106.

Haviluddin, 2011. Memahami penggunaan UML (*Unified Modeling Language*). *Jurnal Informatika Mulawarman*, 6(1): 5-6.

[Kementan] Kementrian Pertanian. 2015. Renstra kementan tahun 2015-2019 [Internet]. [diunduh 2017 Mei 13]. Tersedia pada: http://www.pertanian.go.id/file/RENSTRA\_2015-2019.pdf

Kurniawan E. 2014. Implementasi REST Web service untuk *sales order* dan *sales*

*tracking* berbasis *mobile. Jurnal EKSIS.* 7(1): 1-12.

Oktaviani N, Hutrianto. 2016. *Extreme programming* sebagai metode pengembangan e-keuangan pada pondok pesantren qodratullah. *Jurnal Ilmia MATRIK.* 18(2):165-168.

Pressman RS. 2010. *Software Engineering : A Practitioner's Approach* *Ed ke-7.* Boston (US): Mc Graw Hill.

Rismanto R, Arhandi PP , Prasetyo A. 2016. Rancang bangun aplikasi ujian *online real time* dengan menggunakan aristektur *mean*. *Jurnal Teknologi Informasi.* 7(2): 150-151.

Rohman FN. 2015. Pengembangan aplikasi web pengolah data nilai lomba baris berbaris menggunakan metodologi *Extreme Programming* [skripsi]. Yogyakarta (ID): Universitas Negeri Yogyakarta.

Saputra ZR. 2015. Aplikasi SMS *Center* untuk informasi harga komoditi hasil pertanian Kabupaten Ogan Ilir. *Jurnal SIGMATA*. 4(2): 3-6.

Setiady H. 2013. Sistem informasi pemesanan dan penjualan berbasis web pada dewi florist [skripsi]. Palembang (ID): Sekolah Tinggi Ilmu Manajemen Informatika dan Komputer *Global Information* Multi Data Palembang

Setiana D. 2016. Pengembangan modul pelaporan harga komoditas pertanian pada sisi pengguna admin dan pemerintah 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.

Undang-Undang Republik Indonesia Nomor 8 Tahun 1999 Tentang Perlindungan Konsumen. [Internet]. [diunduh 2017 Jan 02]. Tersedia pada: https://www.kontras.org/uu\_ri\_ham/UU%20Nomor%208%20Tahun%201999%20tentang%20Perlindungan%20Konsumen.pdf

Wihono A. 2009. Analisis volatilitas harga sayuran di Pasar Induk Kramat Jati [skripsi]. Bogor (ID): Institut Pertanian Bogor.