**NUGROHO AGUNG CAHYONO**

# **PENGEMBANGAN MODUL PELAPORAN HARGA KOMODITAS PERTANIAN UNTUK MASYARAKAT, PEDAGANG DAN PEMERINTAH MENGGUNAKAN REST API**

**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 Masyarakat 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, 2017

*Nugroho Agung Cahyono*

NIM G64130039

**ABSTRAK**

NUGROHO AGUNG CAHYONO. Pengembangan Modul Pelaporan Harga Komoditas Pertanian untuk Masyarakat, Pedagang dan Pemerintah Menggunakan REST API. Dibimbing oleh DEAN APRIANA RAMADHAN.

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, *database* yang digunakan yaitu MySQL. Komoditas pertanian yang digunakan pada penelitian tersebut terdiri dari beras, daging sapi, cabai, kedelai, dan bawang merah. *Database*  yang akan digunakan dalam penelitian ini yaitu *non-relational database*. Metode yang digunakan yaitu metode *Extreme Programming* yang merupakan bagian dari *Agile Software Engineering.*

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

**ABSTRACT**

NUGROHO AGUNG CAHYONO. Development of Agricultural Commodity Price Reporting Module for Communities, Traders and Government Using REST API. Supervised by DEAN APRIANA RAMADHAN.

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.

**NUGROHO AGUNG CAHYONO**

**DEPARTEMEN ILMU KOMPUTER**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**INSTITUT PERTANIAN BOGOR**

**BOGOR**

**2017**

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

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

Penguji:

1. Penguji 1
2. Penguji 2

Judul Skripsi : Pengembangan Modul Pelaporan Harga Komoditas Pertanian untuk Masyarakat, Pedagang dan Pemerintah Menggunakan REST API

Nama : Nugroho Agung Cahyono

NIM : G64130039

Disetujui oleh

Dean Apriana Ramadhan, SKomp MKom

Pembimbing

Diketahui oleh

Dr Ir Agus Buono, MSi MKom

Ketua Departemen

Tanggal Lulus:

**PRAKATA**

Puji dan syukur penulis panjatkan kepada Allah *subhanahu wa ta’ala* atas segala karunia-Nya sehingga tugas akhir ini berhasil diselesaikan. Judul yang dipilih dalam penelitian yang dilakukan sejak Februari 2017 ialah Pengembangan Modul Pelaporan Harga Komoditas Pertanian untuk Masyarakat, Pedagang dan Pemerintah Menggunakan REST API. Penelitian ini dilaksanakan di Departemen Ilmu Komputer FMIPA IPB.

Penulis menyadari bahwa selama mengerjakan tugas akhir ini mengalami berbagai kendala. Namun, berkat bantuan, bimbingan, dan dukungan dari berbagai pihak kendala yang dihadapi berhasil diselesaikan. Oleh karena itu penulis ingin mengucapkan terima kasih kepada:

1. Orang tua dan adik penulis yang memberi doa, semangat, dan dukungan moril maupun materiil.
2. Bapak Dean Apriana Ramadhan, SKom MKom selaku dosen pembimbing yang telah sabar, ikhlas, meluangka waktu, tenaga dan pikiran dalam memberi bimbingan, motivasi dan saran selama pengerjaan tugas akhir.
3. Teman-teman satu bimbingan, yaitu Fiqih Nur R, Ryan Baskara, dan Irfan Rafii yang memberikan saran, bantuan dan dukungan selama penelitian berlangsung.
4. Staff dan pengajar Departemen Ilmu Komputer IPB.
5. Teman-teman Sonic angkatan 50 Departemen Ilmu Komputer IPB.
6. Semua pihak yang terlibat dalam menyelesaikan tugas akhir ini baik secara langsung maupun tidak langsung yang tidak dapat penulis sebutkan satu per satu.

Bogor, 2016

*Nugroho Agung Cahyono*

**DAFTAR ISI**

DAFTAR TABEL vi

DAFTAR GAMBAR vi

DAFTAR LAMPIRAN vi

**DAFTAR TABEL**

**DAFTAR GAMBAR**

**DAFTAR LAMPIRAN**

**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. Pada tahun 2014 sektor pertanian menyerap angkatan kerja sebanyak 35.76 juta jiwa atau 30.2% dari keseluruhan angkatan kerja nasional. Selama periode 2010-2014 rata-rata kontribusi petanian terhadap PDB mencapai 10.26% dengan pertumbuhan 3.90% (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, 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). Pengguna internet di Indonesia berdasarkan survey APJII yaitu 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 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). Pada sistem tersebut perlu pengembangan token OAuth menjadi token yang dinamis untuk keamanan data (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) yaitu 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. Modul pelaporan harga komoditas 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 keluaran yang dihasilkan berupa format JSONyang fleksibel dan dapat diakses oleh berbagai *platform,* baik web maupun *mobile*. Penelitian ini menggunakan *non-relational database*.

**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.

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.

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.

**TINJAUAN PUSTAKA**

**JSON(*Java Script Object Notation*)**

JSON (*Java Script Object Notation*) adalah format pertukaran data (*lightweight data-interchange format*), mudah diterjemahkan dan dibuat (*generate*) oleh komputer, serta mudah dibaca dan ditulis oleh manusia. JSON merupakan format teks yang tidak bergantung pada bahasa pemprograman apapun. JSON dirancang untuk memudahkan pertukaran data dan merupakan perluasan dari fungsi-fungsi *Javascript* (Herdiana 2014)*.*

**REST API**

*REST* (*Representational State Transfer*) merupakan sebuah arsitek untuk aplikasi *web services*, dirancang untuk menggunakan protokol HTTP sebagai penghubung komunikasi antara mesin dan mesin, tanpa menggunakan mekanisme yang kompleks. Perintah HTTP yang bisa digunakan adalah fungsi GET, POST, PUT, DELETE. *REST* bersifat *stateless,* yang berarti setiap *request* yang diterima diproses secara *independent*. Format data yang digunakan dalam *REST* adalah *JavaScript Object Notation* (JSON) atau XML (Kurniawan 2014).

**MongoDB**

MongoDB merupaka salah satu jenis *database non-relational* dengan tipe *document store database*. MongoDB dapat digunakan untuk penyimpanan data yang besar dan *high performance* (Ashana 2015). Format data JSON yang digunakan oleh MongoDB memiliki skema pertukaran data yang dinamis sehingga memudahkan dalam mengintegrasikan data dengan lebih mudah dan cepat (Rismanto 2016).

**Express.js**

Express.js merupakan *framework* dari Node.js yang minimalis dan fleksibel yang menyediakan fitur – fitur untuk mendukung aplikasi web dan *mobile*. Secara *de facto* merupakan standar *server framework* untuk Node.js (Rismanto 2016). *Framwork* Express.js menyediakan fitur yang dapat ditambahkan melalui *plugin*.

***Roadmap* Penelitian**

*Roadmap* merupakan perencanaan dari seluruh pekerjaan yang akan dikerjakan secara detail dan terperinci dalam jangka waktu tertentu. Pembuatan *roadmap* agar tujuan yang akan dicapai selesai sesuai target waktu yang telah ditentukan. Tiga tahapan perencanaan *roadmap* yang akan dilakukan oleh tim peneliti e-*Goverment* bidang pertanian pada Laboratorium *Software Engineering and Information Science* (SEInS) Departemen Ilmu Komputer FMIPA IPB dalam rentang waktu tahun 2016 – 2020 yaitu *big data analysis*, *product development*, dan ICT *literacy.* Tiga tahapan tersebut digambarkan secara detail pada Gambar 1.



Gambar 1 *Roadmap* penelitian *e*-*Government* bidang pertanian

Penelitian ini merupakan pelaksanaan *roadmap* pada tahap *product development* bagian *mobile* and *web back end development.* Fokus penelitian ini pada bagian *back end* yang akan menghasilkan API. Fungsi API untuk mengatur dan menjadi sumber aliran data yang dibutuhkan oleh *platform* berbasis web maupun *mobile*. Pengembangan *back end* menggunakan Express.js yang merupakan *framework* Node.js. *Database* yang digunakan yaitu *databae non-relational.*

**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*). Data yang digunakan salah satunya yaitu data komoditas dapat dilihat secara detail pada Tabel 1.

Tabel 1 Data laporan harga

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| komoditas\_id | name | satuan | Harga | datePost | last\_update |
| 10 | Beras | Liter | 10000 | 05 April 2017 | 13 April 2015 |

**Arsitektur Penelitian**

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

Gambar 2 Arsitektur Penelitian

**Tahapan Penelitian**

Metode yang digunakan yaitu *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 (Widodo 2008). Tahapan dalam *Extreme Programming* yaitu *planning, design, coding,* dan *testing* dapat dilihat pada Gambar 3*.*

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

Gambar 3 Tahapan pada *Extreme Programming* (Pressman 2010)

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

Perencanaan *requirement* yang diperlukan yang akan dibuat menjadi modul-modul dan fungsi-fungsi pada sistem. Membuat *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 kabapitilas baru mengenai sebuah fitur pada sebuah sistem (Cohn 2014). *Use case* merupakan deskripsi fungsi dari sebuah sistem berdasarkan perseptif pengguna (Basuki 2011). Iterasi yang akan dilakukan selama penelitian sebanyak tiga kali.

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

Perancangan dilakukan secara berkelanjutan selama proses pengembangan. Perancangan dibuat sederhana dan sesuai dengan rencana iterasi. Perancangan yang dilakukan mencakup pembuatan *class, responsibilities,* dan *collaboration* (CRC) *cards* dan *sequence diagram.* *Class* diagramadalah gambaran struktur sistem berdasarkan dari pendefinisian *class-class* yang akan dibuat dalam membangun sistem (Setiady 2013).Selama proses desain jika mengalami kesulitan dilakukan s*pike solution prototype* yang merupakan sebuah solusi 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. Pada tahap pengkodean melakukan *continuous integration,* yaitu apabila terjadi perubahan *requirement* selama proses pengembangan maka harus segera diintegrasikan (Widodo 2008). 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 yatu Fiqih Nur Ramadhan.

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

Penyusunan *acceptance* *testing* yang berdasarkan fungsi atau fitur yang diperoleh dari *user story* pada proses perencanaan. Pengujian dilakukan secara *black-box* menggunakan *Postman.* Menguji dengan perintah HTTP menggunakan fungsi POST dan GET untuk memasukkan data ke dalam *database* dan mendapatkan data dari *database.* Keluaran data yang diperoleh dalam format JSON yang sesuai dengan masukan. Pengujian yang baik menghasilkan *bugs* yang sedikit.

**Lingkungan Pengembangan**

Spesifikasi yang digunakan pada penelitian ini, perangkat keras dan perangkat lunak sebagai berikut:

Perangkat keras pengembangan:

* *Processor* Intel® Core™ i3-3217u
* RAM 4 GB
* 500 GB HDD
* VGA Ge Force 740M 2 Gb.

Perangkat lunak pengembangan:

* Sistem operasi : Linux Ubuntu 16.04 LTS
* *Text editor* : Visual Studio Code
* DBMS : MongoDB
* Bahasa pemrograman : javascript
* *Platform* : Node.js
* *Framework* : Express.js
* *Test* API : Postman

**HASIL DAN PEMBAHASAN**

**Tahapan Penelitian**

Proses pengembangan API dimulai pada bulan Februari 2017 dengan menggunakan arsitektur yang telah dikembangkan oleh tim dan menggunakan metode *Extreme Programming*. Iterasi yang dilakukan sebanyak tiga kali. 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*, *testing*. Keempat tahap tersebut dilakukan pengulangan sebanyak tiga kali.

**Iterasi 1**

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

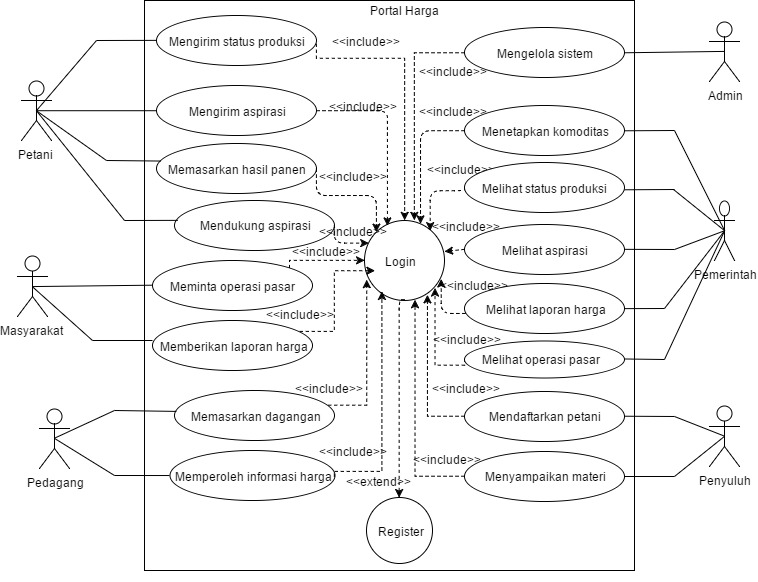
Perencanaan awal dilakukan oleh tim untuk menentukan *stakeholder* dan *role* dari setiap *stakeholder*. Setelah dilakukan diskusi dengan tim, *stakeholder yang terlibat* di dalam sistem ada enam, yaitu *admin*, pemerintah, penyuluh, petani, masyarakat, dan pedagang. *Role* setiap *stakeholder* sebagai 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*.*

Membuat *user story* yang bertujuan 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 data 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. |

Membuat *use case* digramuntuk 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*. *Use case diagram* secara detail dapat dilihat pada gambar 3.



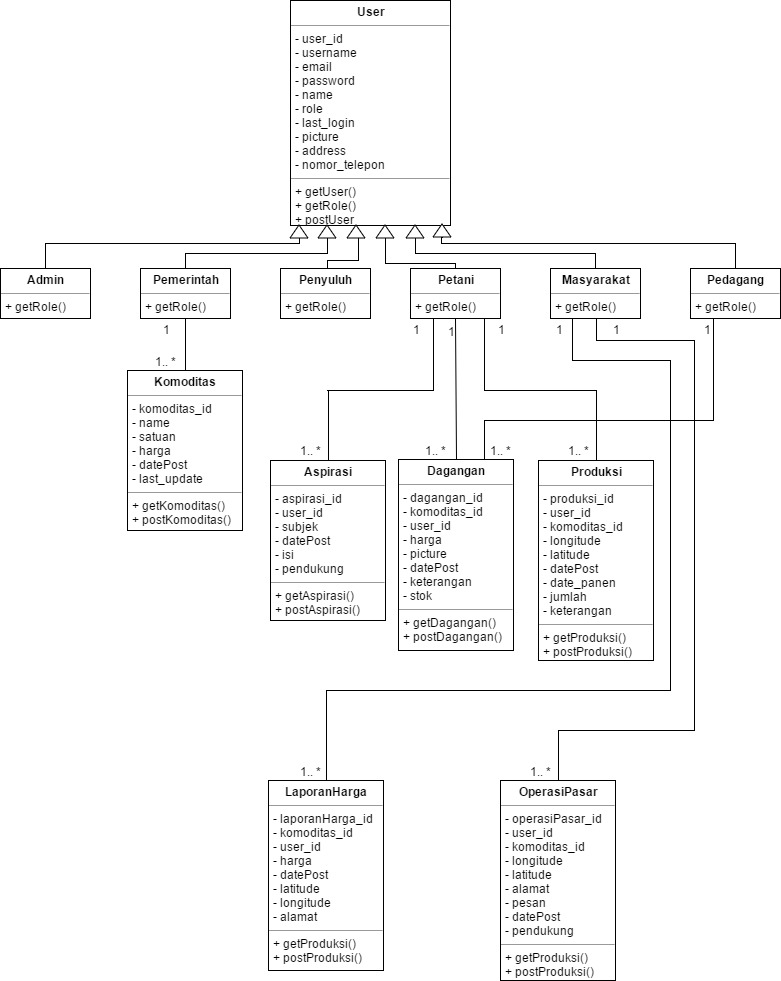
Gambar 3 *Use case diagram* iterasi pertama

Membuat 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 yang akan dibuat fungsi dasar yaitu CRUD.

* 1. **Desain (Design)**

Pada tahap desain dilakukan perancangan dan pembuatan *class* diagram*.* Class diagram memberikan gambaran tentang sistem dan mempermudah dalam pengembangan sistem. *Class diagram* diimplementasikan pada *non-relational database* mongoDB dan setiap *class* yang dibuat akan disimpan dalam sebuah *collections* di dalam *database*. Gambar daftar *collections*  dapat dilihat pada gambar 4. 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 diagram* secara detail dapat dilihat pada gambar 5.

Gambar 4 Daftar *collections* *database*



Gambar 5 *Class diagram* iterasi pertama

*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.* Turunan tersebut tergantung dari *role* yang dimiliki oleh setiap *user.* Suatu *class* tidak selalu bisa diakses oleh *class* yang lain. Sebagai contoh *class* laporanHarga dan *class* operasiPasar dapat diakses oleh *class* masyarakat namun tidak bisa diakses oleh *class penyuluh*.

Pada tahan desain juga dibuat *sequence* diagramyang merupakan gambaran interaksi dan komunikasi diantara objek-objek. *Sequence* diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu.Salah satu *sequence* diagram yaitu fungsi tambah laporan pada modul laporan harga yang dapat dilihat pada gambar 6.

Gambar 6 *Sequece* diagram fungsi tambah laporan

* 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 yaitu 5000. Penggunaan *method* POST dan GET diatur di dalam *route*, sebagai contoh *route* modul laporan harga dapat dilihat pada gambar 7. *Method* POST digunakan untuk memasukkan data ke dalam *database*, sedangkan *method* GET untuk memperoleh data dari database. Pada *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>. Potongan *code* REST API fungsi menambah laporan harga menggunakan *method* POST dapat dilihat pada gambar 8. Contoh REST API dengan *method* GET yaitu dengan mengakses URL <https://ph.yippytech.com:5000/laporanHarga/get>. Potongan *code* REST API fungsi mengambil data laporan harga menggunakan *method* GET dapat dilihat pada gambar 9. 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 (“{}”). Contoh keluaran dengan *method* GET pada fungsi mengambil laporan harga dari database dapat dilihat pada gambar 10 dengan format keluaran JSON yang dihasilkan terdiri dari status, data, dan token.

Gambar 7 *Route* modul laporan harga

Gambar 8 Potongan *code* fungsi memasukkan data laporan harga menggunakan *method* POST

Gambar 9 Potongan *code* fungsi mengambil data laporan harga menggunakan *method* GET

Gambar 10 Format keluaran JSON *array* dan JSON *object*

Token yang digunakan dalam sistem yaitu 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*. Sedangkan untuk *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* yaitu 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 yaitu pengujian token, mengecek ketersediaan token dan memvalidasi token yang diberikan. Pengujian selanjutnya yaitu mengecek dan memvalidasi *role* yang berasal dari hasil *decode* token. *Role* digunakan supaya dapat mengakses suatu fungsi tertentu. Pengujian terakhir berhubungan dengan *database*, yaitu mengambil data dari *database* jika data yang diminta tidak tersedia di *database* atau kosong, gagal menyimpan data ke dalam *database,* dan parameter yang diberikan tidak sesuai yang ada di *database*. Pengujian modul laporan harga secara detail dapat dilihat pada tabel 3.

Tabel 3 Pengujian REST API modul laporan harga pada iterasi pertama

**NANTI DIUBAH**

|  |  |  |
| --- | --- | --- |
| *Actor* | Fungsi | Status Pengujian |
| Semua *actor* | GET *Check* Token | Berhasil |
| GET *Check Role* | Berhasil |
| Masyarakat | CRUD Laporan Harga | Berhasil |
|  | CRUD Operasi Pasar | Berhasil |
| Pemerintah | CRUD Komoditas | Berhasil |

**Iterasi 2**

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

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 *database* 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 yaitu *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 *passwaord*.

***Planning iterasi ke tiga nanti***

Berdasarkan analisis *requirement* pada iterasi kedua terdapat dua modul baru untuk mendukung sistem yang akan dikembangkan, yaitu modul lokasi dan email. Modul lokasi bertujuan untuk memberikan pilihan lokasi kepada pengguna. Validasi *account* dikirim melalui email dan jika divalidasi maka tidak bisa melakukan *login*. *Use case* diagramsecara detail dapat dilihat pada gambar 8.

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. Pada *class* laporan harga terdapat dua fungsi baru, yaitu fungsi laporanku dan getDayLaporan. *Class* email hanya terdapat satu fungsi yaitu *forgetPassword*. *Class* digram setelah semua modul dan fungsi disatukan dapat dilihat pada gambar 11.

Gambar 11 *Class* diagram iterasi kedua

Salah satu *sequence* diagram yaitu untuk fungsi *forgetPassword* yang dapat dilihat pada gambar 12. //jelaskan isi sequence diagram

Gambar 12 *Sequence* diagram fungsi *forgetPassword*

//masuk ke iterasi ke tigas

Terdapat *class* baru yaitu *class* lokasi dan *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* yang akan mengirimkan email ke pengguna jika pengguna lupa dengan *password*nya dengan memberikan suatu *link* yang disisipi parameter hasil *encode* *username*. Kedua *update* *password* yang merupakan lanjutan dari fungsi *forgetPassword* untuk memperbarui *password* pengguna berdasarkan *username* yang sesuai. Ketiga getValidating merupakan fungsi yang akan mengirim email kepada pengguna ketika pengguna pertama kali mendaftar atau ketika *login* namun belum melakukan validasi *account*. Keempat fungsi validating yaitu memvalidasi *account* dari *link* yang diklik pada email yang diterima pengguna.

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. Potongan *code* mendukung operasi pasar dapat dilihat pada gambar12.

Gambar 12 Potongan *code* mendukung operasi pasar

Iterasi kedua pada modul laporan harga untuk fungsi getDayLaporan dilakukan *time out* selamat 500 *miliseconds*. *Time out* selama 400 *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*. Potongan *code* fungsi getDayLaporan dapat dilihat pada gambar 13.

Gambar 13 Potongan *code* fungsi getDayLaporan

Implementasi modul email menggunakan npm nodemailer yang merupakan gmail Google. Potongan *code* fungsi *forgetPassword* dapat dilihat pada gambar 14. 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 12 karakter.

Gambar 14 Potongan *code* fungsi *forgetPassword*

*Pair programming* dilakukan antara fungsi getValidating email dengan fungsi *login* dan fungsi register yang dapat dilihat pada gambar 10 dan gambar 11. Gambar 11 merupakan fungsi getValidating email.

Gambar 10 Integrasi fungsi getValidating dengan fungsi register

Gambar 11 Integrasi fungsi getValidating dengan fungsi *login*

Pembuatan API untuk modul lokasi menggunakan data lokasi seluruh indonesia dalam bentuk sql. Data sql kemudia diubah menjadi format JSON untuk masing masing tabel menggunakan Phpmyadmin. Data sql terdiri dari lima tabel yaitu, jenis, provinsi, kabupaten, kecamatan, dan kelurahan. Setelah data dalam format JSON kemudian dimasukkan ke dalam *database* mongoDB dengan menggunakan *terminal* untuk masing-masing tabel menjadi *collections* yang bersesuaian. *Code* di *terminal* yang digunakan untuk memasukkan data format JSON ke dalam *database* mongoDB dapat dilihat pada gambar 12.

Gambar 12 *Code* memasukkan data format JSON ke *database* mongoDB

Format keluaran JSON yang diperoleh ketika mengakses API lokasi pada bagian data yaitu *array object* yang terdiri dari id\_lokasi yang dipilih dan nama lokasinya.

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. Hasil pengujian pada iterasi kedua dapat dilihat pada tabel 4.

|  |  |  |
| --- | --- | --- |
| *Actor* | Fungsi REST | Status Pengujian |
| Masyarakat | POST *Vote* operasi pasar | Sukses |
|  | POST *Unvote* operasi pasar | Sukses |
|  | GET Operasiku | Sukses |
|  | GET Pendukung | Sukses |
|  | GET laporanku | Sukses |
|  | GET *Day* laporan | Sukses |
| Semua *Actor* | GET *Forget password* | Sukses |

Tabel 4 Hasil pengujian iterasi kedua

**Iterasi 3**

1. **Pengujian (*Testing*)**
2. **Sfd**
3. **sdfsdfs**