

**PENGEMBANGAN MODUL PELAPORAN HARGA
PADA APLIKASI DIGITAL TANI**

NUGROHO AGUNG CAHYONO



**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 pada Aplikasi Portal Harga adalah benar karya saya dengan arahan dari komisi pembimbing dan belum diajukan dalam bentuk apa pun kepada perguruan tinggi mana pun. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam Daftar Pustaka di bagian akhir skripsi ini.

Dengan ini saya melimpahkan hak cipta dari karya tulis saya kepada Institut Pertanian Bogor.

Bogor, Agustus 2017

Nugroho Agung Cahyono
NIM G64130039

ABSTRAK

NUGROHO AGUNG CAHYONO. Pengembangan Modul Pelaporan Harga pada Aplikasi Digital Tani. Dibimbing oleh DEAN APRIANA RAMADHAN.

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

NUGROHO AGUNG CAHYONO. Development of Price Reporting Module on Digital Tani Applications. Supervised by DEAN APRIANA RAMADHAN.

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

**PENGEMBANGAN MODUL PELAPORAN HARGA
PADA APLIKASI DIGITAL TANI**

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

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

Penguji:

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

Judul Skripsi: Pengembangan Modul Pelaporan Harga pada Aplikasi Digital Tani
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: 10 AUG 2017

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 materiel.
- 2 Bapak Dean Apriana Ramadhan, SKomp MKom selaku dosen pembimbing yang telah sabar, ikhlas, meluangkan 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 Pegawai 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, Agustus 2017

Nugroho Agung Cahyono

DAFTAR ISI

DAFTAR TABEL	viii
DAFTAR GAMBAR	viii
DAFTAR LAMPIRAN	viii
PENDAHULUAN	1
Latar Belakang	1
Perumusan Masalah	2
Tujuan Penelitian	2
Manfaat Penelitian	2
Ruang Lingkup Penelitian	2
TINJAUAN PUSTAKA	3
JSON	3
REST API	3
MongoDB	3
<i>Roadmap</i> Penelitian	3
METODE	4
Data Penelitian	4
Arsitektur Sistem	5
Tahapan Penelitian	5
Lingkungan Pengembangan	7
HASIL DAN PEMBAHASAN	7
Tahapan Penelitian	7
SIMPULAN DAN SARAN	22
Simpulan	22
Saran	22
DAFTAR PUSTAKA	22
LAMPIRAN	27
RIWAYAT HIDUP	55

DAFTAR TABEL

1 Tahapan pengembangan setiap iterasi	7
2 <i>User story</i> masyarakat	8
3 Hasil pengujian iterasi pertama	12
4 Hasil pengujian iterasi kedua	16
5 Hasil pengujian iterasi ketiga	20

DAFTAR GAMBAR

1 <i>Roadmap</i> penelitian <i>e-Government</i> bidang pertanian	4
2 Arsitektur Penelitian	5
3 Tahapan pada <i>Extreme Programming</i> (Pressman 2010)	6
4 <i>Use case diagram</i> iterasi pertama	8
5 <i>CRC cards</i> iterasi pertama	9
6 <i>Class diagram</i> iterasi pertama	9
7 <i>Collections</i> komoditas	10
8 Daftar <i>collections</i> di basis data iterasi pertama	10
9 <i>Unit test</i> menambah komoditas	11
10 Pengujian fungsi melihat laporan harga	12
11 <i>Use case diagram</i> iterasi kedua	13
12 <i>CRC cards</i> iterasi kedua	14
13 <i>Class diagram</i> iterasi kedua	14
14 Potongan <i>code</i> fungsi <i>getDayLaporan</i>	15
15 Pengujian fungsi melihat laporan harga 60 hari sebelumnya	16
16 <i>Use case diagram</i> iterasi ketiga	17
17 <i>CRC cards</i> iterasi ketiga	18
18 <i>Class diagram</i> iterasi ketiga	19
19 <i>Code</i> memasukkan data format JSON ke basis data mongoDB	20
20 Integrasi fungsi <i>getMailVerify</i> dengan fungsi <i>register</i>	20
21 Dokumentasi API di Swaggerhub	21

DAFTAR LAMPIRAN

1 Deskripsi <i>user story</i>	24
2 <i>Sequence diagram</i> iterasi pertama	25
3 <i>Collaboration diagram</i> iterasi pertama	27
4 Daftar <i>collections</i> iterasi pertama	28
5 Kamus data	29
6 <i>Acceptance test criteria</i> iterasi pertama	31
7 <i>Sequence diagram</i> iterasi kedua	34
8 <i>Collaboration diagram</i> iterasi kedua	38
9 <i>Acceptance test criteria</i> iterasi kedua	41
10 <i>Sequence diagram</i> iterasi ketiga	43
11 <i>Collaboration diagram</i> iterasi ketiga	49
12 <i>Acceptance test criteria</i> iterasi ketiga	53

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 tentang sistem informasi pelaporan harga sebelumnya telah dilakukan, di antaranya oleh Saputra (2015) dengan membangun aplikasi *Short Message Service (SMS) center* untuk informasi harga komoditi hasil pertanian Kabupaten Ogan Ilir. Admin pada sistem *SMS center* tersebut memberikan informasi yang dibutuhkan oleh pengguna. Pengguna mengirimkan SMS ke sistem untuk mendapatkan informasi harga komoditas pertanian, lalu sistem akan memberikan balasan SMS sesuai dengan permintaan yang telah dikirim oleh pengguna. SMS yang masuk maupun keluar dikelola oleh admin, kemudian diintegrasikan dengan dinas terkait sesuai dengan harga komoditas pertanian. Penelitian yang dilakukan oleh Ahmad (2015) tentang perancangan aplikasi komoditas pertanian berbasis Android. Pada sistem tersebut komoditas yang digunakan yaitu padi, jagung, kedelai. Pengguna dapat mengakses sistem untuk mendapatkan informasi komoditas, informasi harga, informasi cuaca, dan berita.

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.

TINJAUAN PUSTAKA

JSON

Java Script Object Notation (JSON) adalah format pertukaran data (*lightweight data-interchange format*), mudah diterjemahkan dan dibuat oleh komputer, serta mudah dibaca dan ditulis oleh manusia (JSON 2002). JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun. JSON bersifat interoperabilitas sehingga dapat mengatasi data yang berukuran besar dan untuk berinteraksi dengan berbagai sistem maupun *platform* tanpa batasan akses (JSON 2002). Format data JSON yang digunakan dalam penelitian ini memiliki komponen yang terdiri atas *array*, *boolean*, *number*, *object*, dan *string*.

REST API

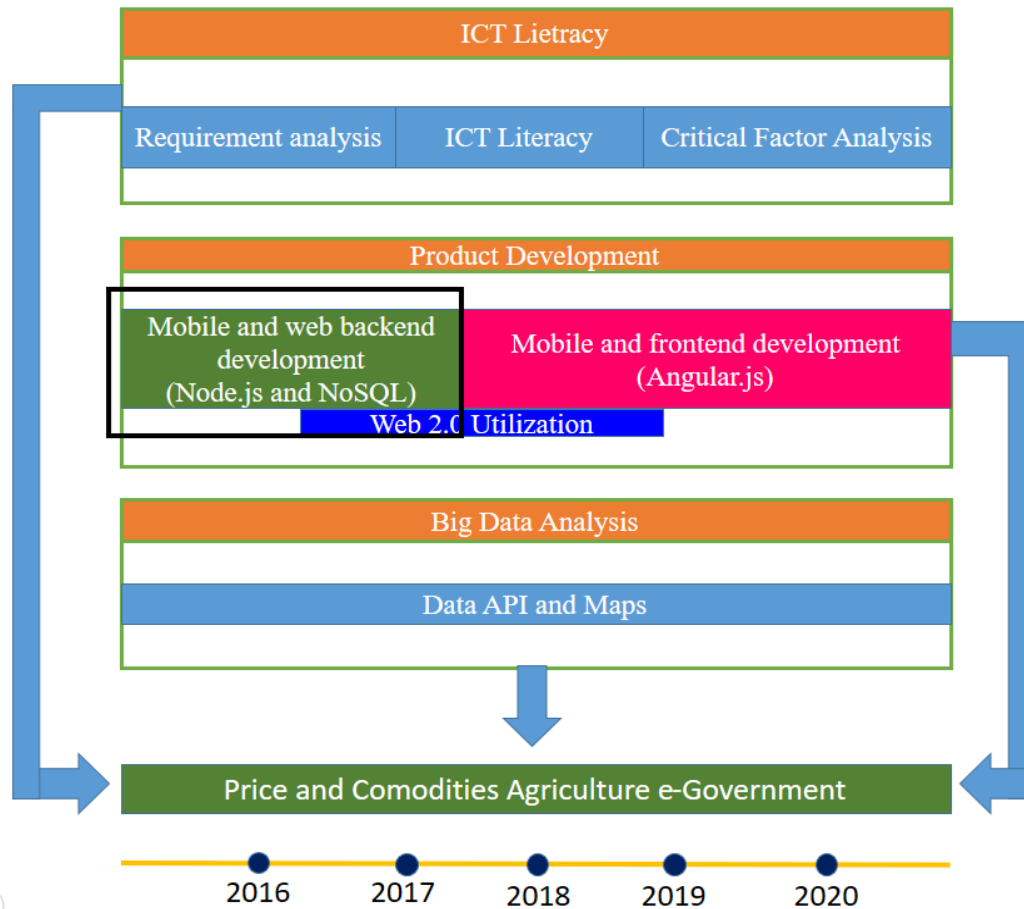
Representational State Transfer (REST) merupakan sebuah arsitek untuk aplikasi *web services*, dirancang untuk menggunakan protokol *Hypertext Transfer Protocol* (HTTP) sebagai penghubung komunikasi antara mesin dan mesin, tanpa menggunakan mekanisme yang kompleks. Perintah HTTP yang bisa digunakan adalah fungsi GET, POST, PUT, DELETE (Pautasso 2008). REST bersifat *stateless*, yang berarti setiap *request* yang diterima diproses secara *independent*. Format data yang digunakan oleh REST pada penelitian ini adalah *JavaScript Object Notation* (JSON).

MongoDB

MongoDB merupakan salah satu jenis basis data non relasional dengan tipe *document store* basis data. MongoDB digunakan dalam penelitian ini untuk menyimpan data yang besar dalam jangka panjang dan *high performance* (MongoDB 2017). Metode *extreme programming* diintegrasikan dengan basis data MongoDB untuk mengoptimalkan data dengan perubahan cepat.

Roadmap Penelitian

Roadmap merupakan perencanaan dari seluruh pekerjaan yang akan dikerjakan secara detil dan terperinci dalam jangka waktu tertentu. Pembuatan *roadmap* bertujuan untuk mencapai target dengan waktu yang telah ditentukan. Tiga tahapan perencanaan *roadmap* yang akan dilakukan oleh tim peneliti e-Government bidang pertanian pada Laboratorium *Software Engineering and Information Science* (SEIS) 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 detil 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, yaitu pada bagian *back end* yang 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. Basis data yang digunakan adalah basis data non relasional.

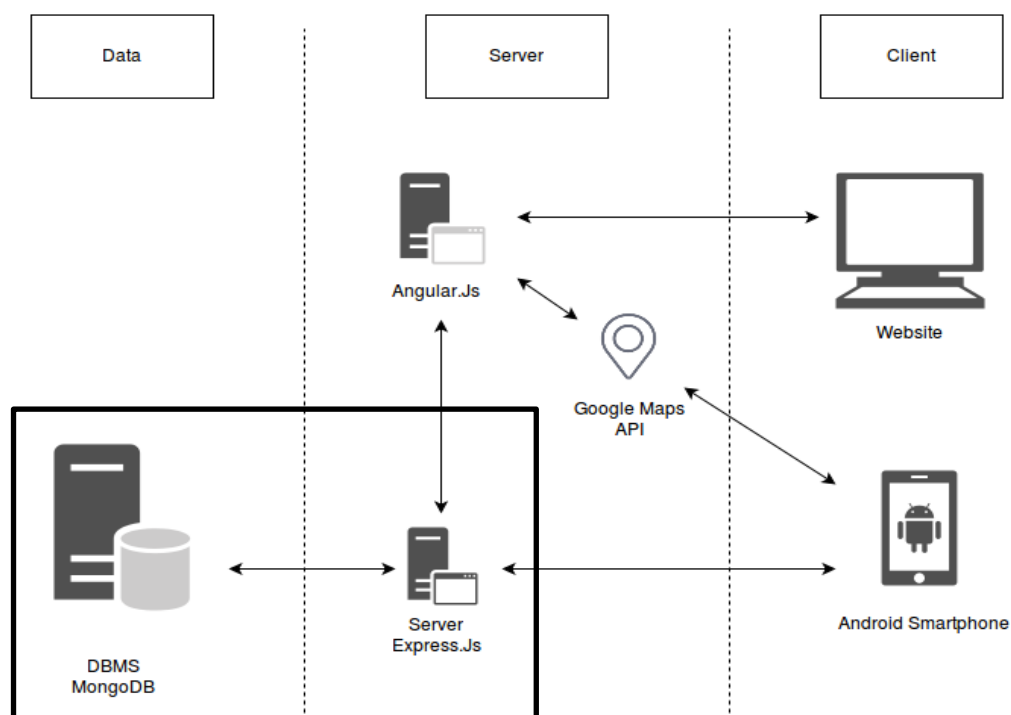
METODE

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.

Arsitektur Sistem

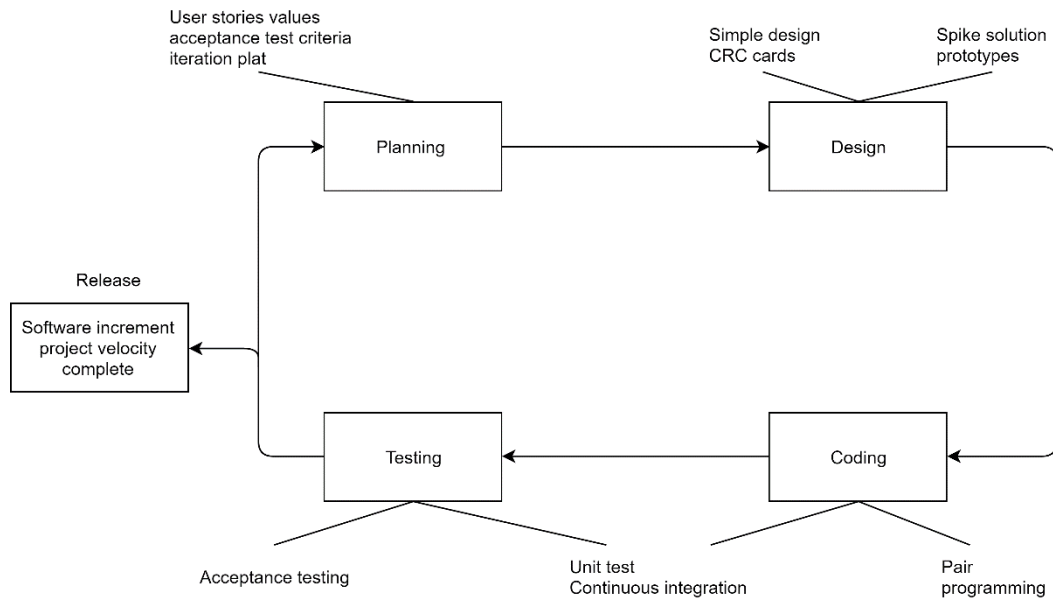
Perancangan arsitektur perangkat lunak dibagi menjadi tiga bagian, yaitu data, *client*, dan *server* (Setiana 2016). Arsitektur perancangan tersebut kemudian dikembangkan dan digunakan selama penelitian. Hasil pengembangan arsitektur penelitian dapat dilihat pada Gambar 2. Bagian data menggunakan basis 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

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 3.



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

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.

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).

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>.

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

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 yang dapat dilihat pada Tabel 1.

Tabel 1 Tahapan pengembangan setiap iterasi

Iterasi	Task	Deskripsi
Pertama	<i>Reengineering</i> modul laporan harga, komoditas, dan operasi pasar	Perubahan basis data relasional MySQL ke basis data non relasional MongoDB.
Kedua	Pengembangan modul <i>email</i>	Pengguna mendapatkan notifikasi lupa <i>password</i> melalui <i>email</i> .
Ketiga	Pengembangan modul lokasi	Lokasi yang terdiri atas provinsi, kabupaten, kecamatan, dan kelurahan.

Iterasi Pertama

Perencanaan (*Planning*)

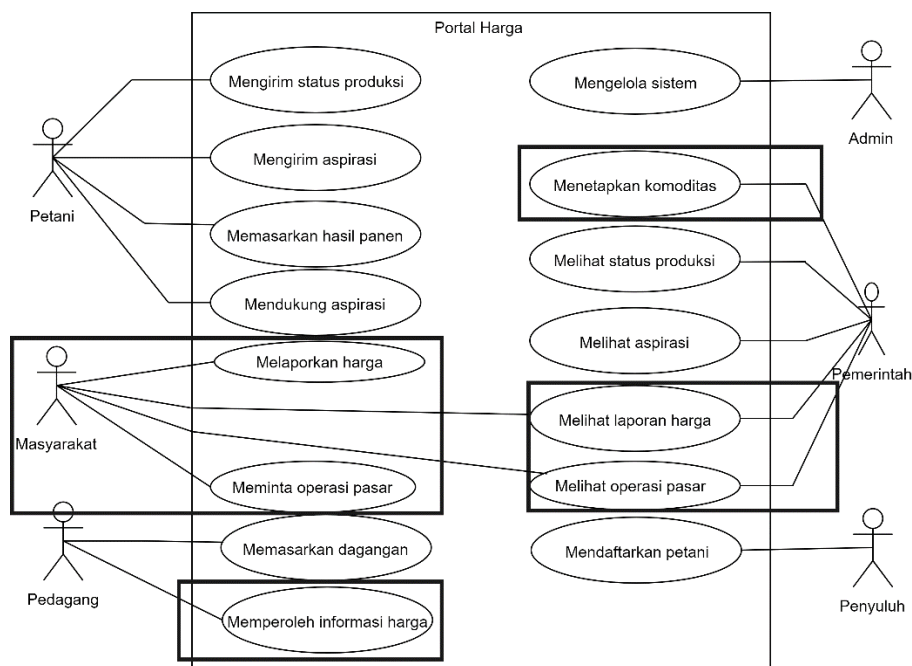
Iterasi pertama melakukan *reengineering* modul dan fungsi–fungsi dasar yang dilakukan oleh Setyatama (2016). Modul yang dikembangkan penulis, yaitu modul komoditas, laporan harga, dan operasi pasar. Modul tersebut diakses oleh *stakeholder* pemerintah, masyarakat, dan pedagang. Fungsi dasar yang akan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

User story dan *use case diagram* dibuat sesuai dengan modul dan *stakeholder* terkait. Salah satu *user story* yang dikembangkan oleh Setyatama (2016), yaitu *user story* masyarakat yang dapat dilihat pada Tabel 2. Detil deskripsi hasil pengembangan *user story* dapat dilihat pada Lampiran 1. *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*.

Tabel 2 *User story* masyarakat

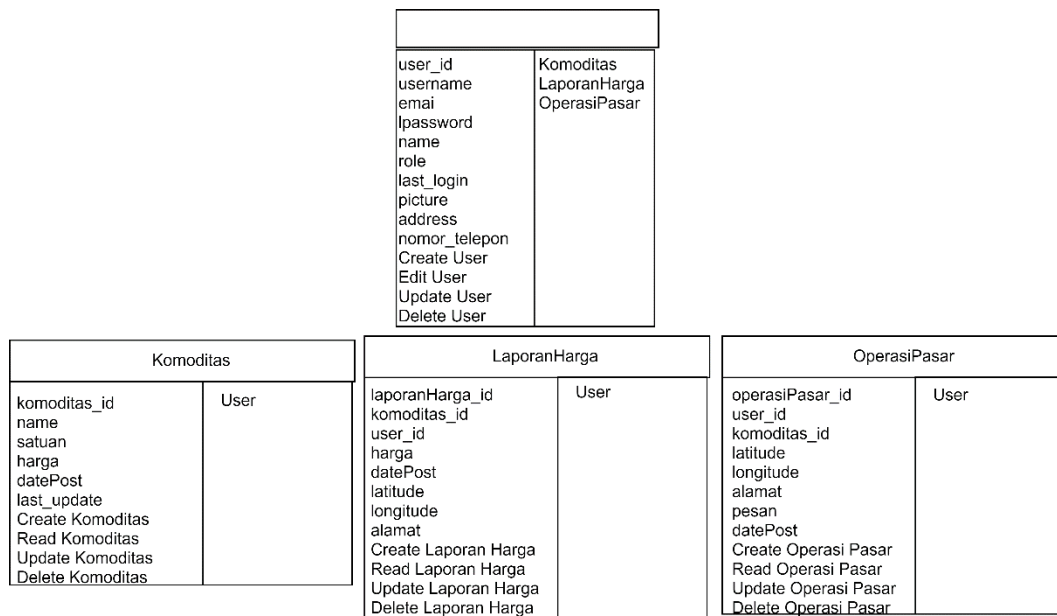
<i>Actor</i>	<i>Task</i>	<i>User story</i>
Masyarakat	Memberikan laporan harga	Pengguna memberikan laporan harga ketika berada pada suatu lokasi pasar. Pengguna memasukkan harga suatu komoditas, sehingga bisa diketahui oleh pengguna lain.
	Melihat laporan harga	Masyarakat memperoleh informasi harga komoditas 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 suatu 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.



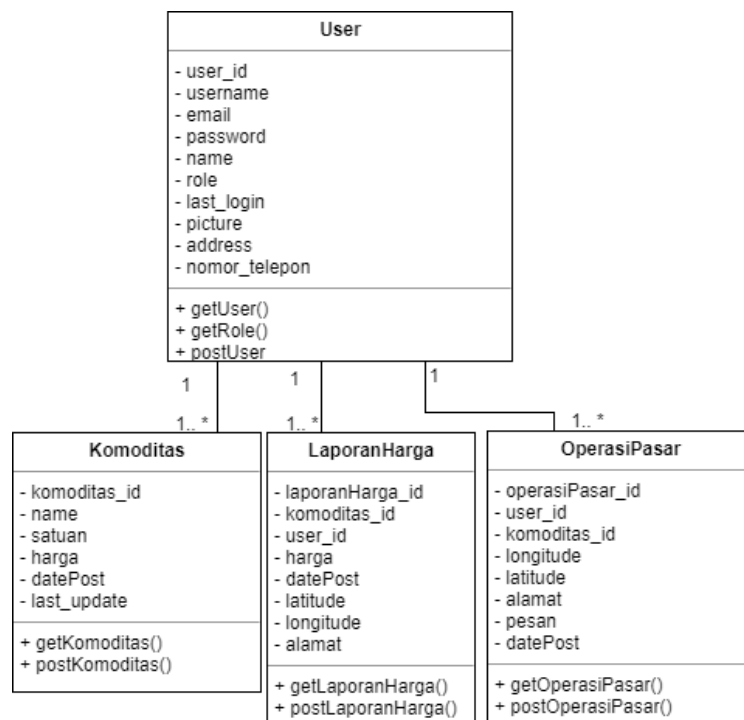
Gambar 4 *Use case diagram* iterasi pertama

Desain (*Design*)

Class Responsibility Collaborator (CRC) cards dirancang menjadi *class diagram* sesuai dengan modul dan *stakeholder* yang sudah ditentukan yang dapat dilihat pada Gambar 5. *CRC cards* memberikan gambaran atribut yang akan dimiliki oleh sebuah *class*, apa saja yang mampu dilakukan oleh sebuah *class*, dan *class* lain yang dapat mengakses suatu *class*.



Gambar 5 *CRC cards* iterasi pertama



Gambar 6 *Class diagram* iterasi pertama

Tahap desain juga melakukan perancangan *sequence diagram* dan *collaboration diagram*. *Sequence diagram* menunjukkan aliran fungsionalitas yang disusun berdasarkan urutan waktu. Detil *sequence diagram* iterasi pertama dapat dilihat pada Lampiran 2. *Collaboration diagram* menunjukkan informasi yang sama persis dengan *sequence diagram*, namun lebih menunjukkan pada hubungan antar *object*. *Collaboration diagram* iterasi pertama dapat dilihat di Lampiran 3.

Class diagram iterasi pertama terdiri atas *class* komoditas, laporan harga, dan operasi pasar. *Class diagram* iterasi pertama dapat dilihat pada Gambar 6. *Collections* komoditas memiliki *field* *komoditas_id*, *name*, *satuan*, *harga*, *datePost*, dan *last_update* yang dapat dilihat pada Gambar 7. *Komoditas_id* merupakan *field primary key* yang bersifat *auto increment*. Atribut *name*, *satuan*, dan *harga* merupakan nama komoditas, satuan komoditas, dan harga komoditas yang berasal dari masukan yang dilakukan oleh pemerintah. *DatePost* adalah waktu saat memasukkan data. *Last_update* adalah waktu terakhir saat memperbarui data komoditas. Daftar *collections* iterasi pertama dapat dilihat pada Lampiran 4.

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). Gambar daftar *collections* iterasi pertama dapat dilihat pada Gambar 8.

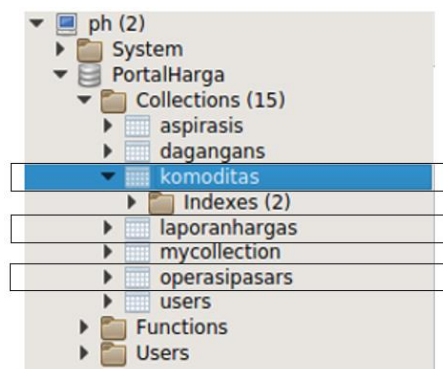
Penentuan atribut pada sebuah *class* menggunakan konsep *Big-O*. Hal tersebut bertujuan untuk mengatasi segala kemungkinan atribut yang dimiliki oleh *object* termasuk turunan dari sebuah *class*. *Class user* atributnya menggunakan konsep *Big-O* untuk menampung segala kemungkinan atribut yang dimiliki oleh setiap jenis *user*. Detil penjelasan atribut pada setiap *collections* dapat dilihat di kamus data pada Lampiran 5.

```

5 | var connection = mongoose.createConnection("mongodb://localhost/PortalHarga");
6 | autoIncrement.initialize(connection);
7 | var komoditasModel = new Schema({
8 |   name:String,
9 |   satuan:String,
10 |   harga:Number,
11 |   datePost:Number,
12 |   last_update:Number
13 | });
14 | komoditasModel.plugin(autoIncrement.plugin,{model:'komoditas',field:'komoditas_id',startAt:1});
15 | module.exports = mongoose.model("komoditas",komoditasModel);

```

Gambar 7 *Collections* komoditas



Gambar 8 Daftar *collections* di basis data iterasi pertama

Salah satu pengujian pada modul laporan harga, yaitu pengujian fungsi melihat laporan harga yang dapat dilihat pada Gambar 10. Pengujian fungsi melihat laporan harga dengan mengakses URL <https://ph.yippytech.com:5000/laporanHarga/get> menggunakan *method* GET. Format keluaran berupa JSON yang terdiri atas status, *message*, data, dan token. Data berisi semua laporan harga yang masing-masing laporan harga terdiri atas laporanHarga_id, datePost, user_id, komoditas_id, harga, *latitude*, *longitude*, alamat, namaKomoditas, satuan, dan nama pengirim laporan harga.

```

1 {
2   "status": 200,
3   "message": "sukses ambil semua laporan harga",
4   "data": [
5     {
6       "laporanHarga_id": 82,
7       "datePost": "2017-08-03",
8       "user_id": 17,
9       "komoditas_id": 10,
10      "harga": 15000,
11      "latitude": -6.5583933,
12      "longitude": 106.7321767,
13      "alamat": "Jl. Babakan Tengah No.64, Babakan, Dramaga, Bogor, Jawa Barat 16680, Indonesia",
14      "namaKomoditas": "Beras",
15      "satuan": "Kg",
16      "nama": "Ryan Baskaraa"
17    },
18    {
19      "laporanHarga_id": 81,
20      "datePost": "2017-07-25",
21      "user_id": 16,
22      "komoditas_id": 9,
23      "harga": 35000,
24      "latitude": -6.5582178,
25      "longitude": 106.7311765,
26      "alamat": "Jl. Meranti, Babakan, Dramaga, Bogor, Jawa Barat 16680, Indonesia",
27      "namaKomoditas": "Cabai",
28      "satuan": "Kg",
29      "nama": "Ryan Baskara"
30    },
31    {
32      "laporanHarga_id": 80,
33      "datePost": "2017-07-25",
34      "user_id": 13,

```

Gambar 10 Pengujian fungsi melihat laporan harga

Tabel 3 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
	POST	Menghapus laporan harga	laporanHarga/delete	Berhasil
Operasi pasar	POST	Menambah operasi pasar	operasiPasar/add	Berhasil
	GET	Melihat operasi pasar	operasiPasar/get	Berhasil
	POST	Mengubah operasi pasar	operasiPasar/update	Berhasil

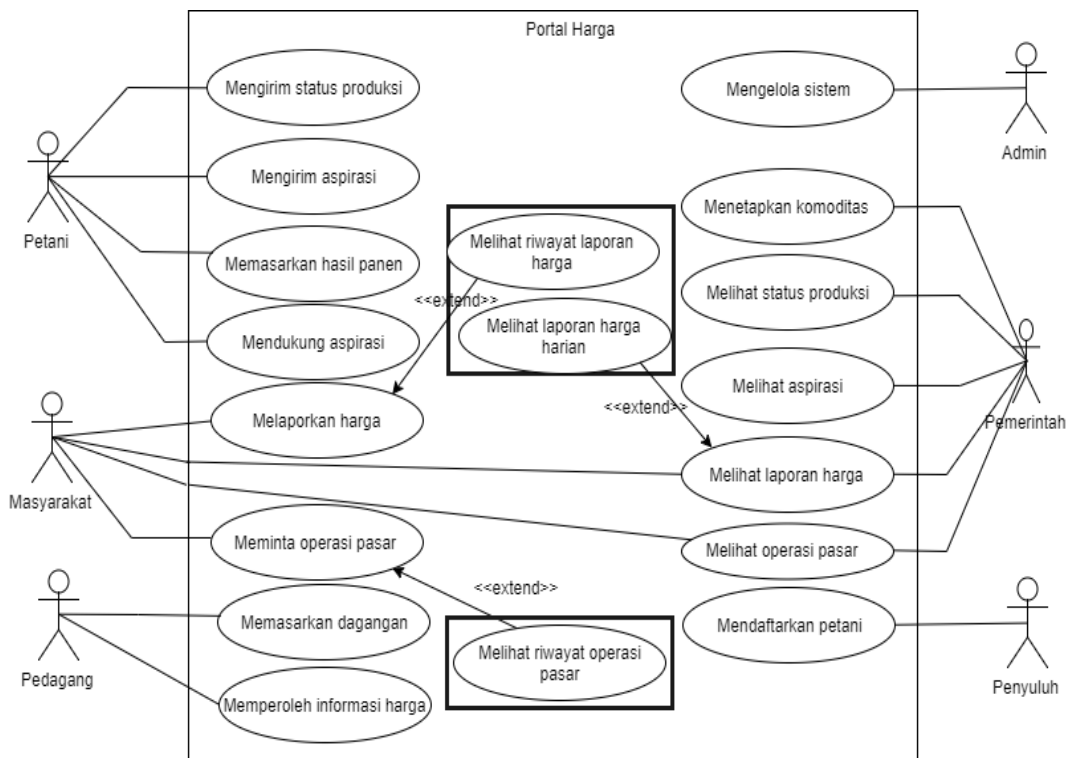
Tabel 3 Lanjutan

<i>Class</i>	<i>Method</i>	Fungsi	URL	Status
Operasi pasar	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. Perubahan *use case diagram* pada iterasi kedua dapat dilihat pada Gambar 11. Fungsi tiap modul yang dikembangkan di iterasi kedua pada *use case diagram* diberi kotak hitam.

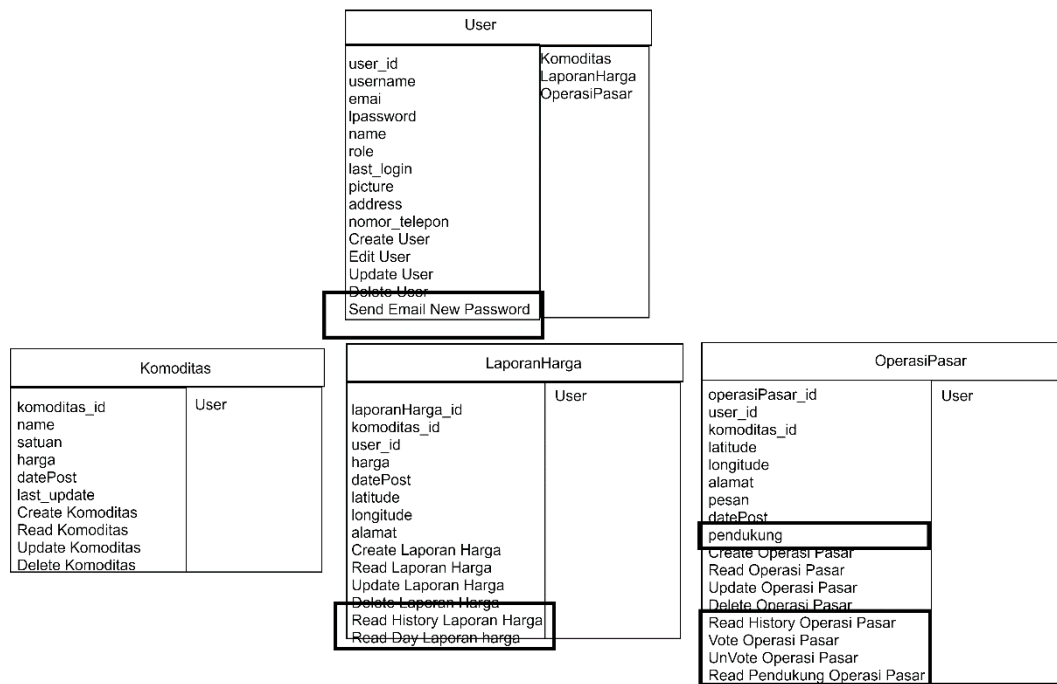


Gambar 11 Use case diagram iterasi kedua

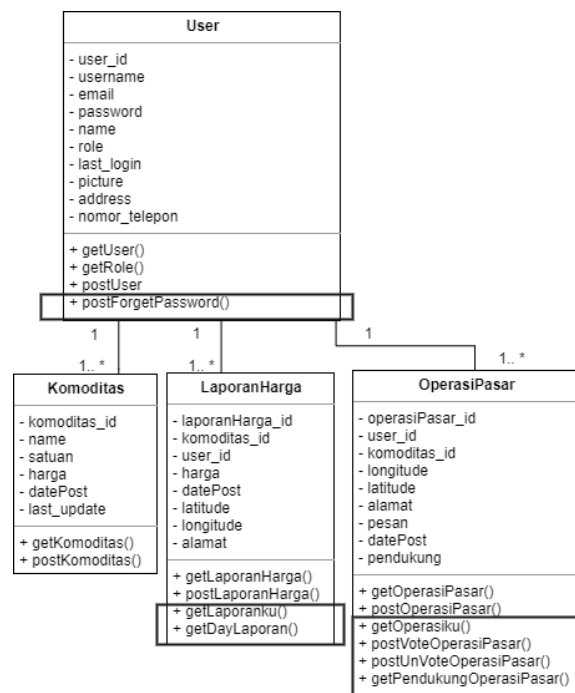
Desain (*Design*)

Class Responsibility Collaborator (CRC) cards dirancang menjadi *class diagram* sesuai dengan modul pada tahap perencanaan yang dapat dilihat pada Gambar 12. Perancangan *class diagram* sesuai dengan *CRC cards*. *Class diagram* iterasi kedua dapat dilihat pada Gambar 13. Pembuatan *sequence diagram* disesuaikan dengan fungsi yang ditambahkan pada *class diagram*. Detil *sequence diagram* iterasi kedua dapat dilihat pada Lampiran 7. *Collaboration diagram* untuk

menunjukkan hubungan antar *object* yang ada pada *sequence diagram* dapat dilihat pada Lampiran 8.



Gambar 12 CRC cards iterasi kedua



Gambar 13 Class diagram iterasi kedua

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.

Tahap pengodean menggunakan *framework* Express.js yang bersifat *asynchronous*. *Asynchronous* dapat diatasi dengan menggunakan *time out* yang dapat dilihat pada Gambar 14. Salah satu fungsi yang menggunakan *time out*, yaitu fungsi *getDayLaporan* dengan *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 dalam *array*.

```
//mengambil laporanHarga beberapa hari yang lalu, komoditasnya apa aja selama hari itu
var dayLaporan = function(req,res){
  //ambil semua laporanHarga di sorting sesuai dengan tanggal post
  laporanHarga.find({}, '-_id - __v', {sort: {datePost: -1}}, function(err, all){
    if(all==null){
      res.json({status:204,message:err,data:"",token:req.token});
    }else{
      //tanggal sekarang
      var dateNow = new Date();
      //tanggal sekarang di kurangi hari yang diinginkan, hari nya
      dateNow.setDate(dateNow.getDate() - req.params.day);
      //hari yang diinginkan dalam format, hari, tanggal, bulan, dan tahun
      var getDate = dateFormat(dateNow, "dddd , mmmm dS , yyyy");
      //console.log(getDate);
      //buat variabel parsing yang akan menerima laporanHarga_id pada hari itu
      var parsing = [];
      var number = [];
      var counter = 0;

      for(var i=0;i<all.length;i++){
        if(dateFormat(all[i].datePost, "dddd , mmmm dS , yyyy")==getDate){
          number.push(all[i].laporanHarga_id);
        }
      }
      //time out 300 miliseconds
      setTimeout(function () {
        for(var i=0;i<number.length;i++){
          laporanHarga.findOne({laporanHarga_id:number[i]}, '-_id - __v').lean().exec(function(err, laporan){
            komoditas.findOne({komoditas_id:laporan.komoditas_id}).exec(function(err, komoditas){
              user.findOne({user_id:laporan.user_id}, function(err, masyarakat){
                laporan.namaKomoditas=komoditas.name;
              }
            )
          )
        }
      }, 300)
    }
  })
}
```

Gambar 14 Potongan *code* fungsi *getDayLaporan*

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. Pengujian pada iterasi kedua dilakukan sesuai dengan *acceptance test criteria* iterasi kedua yang dapat dilihat pada Lampiran 9. 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 4.

Salah satu pengujian pada modul laporan harga, yaitu pengujian fungsi melihat laporan harga beberapa hari sebelumnya yang dapat dilihat pada Gambar 15. Contoh pengujian, yaitu melihat laporan harga 61 hari sebelumnya dengan mengakses URL <https://ph.yippytech.com:5000/laporanHarga/get/day/61> menggunakan *method* GET. Angka 60 merupakan sebuah parameter berapa hari sebelumnya yang dapat diganti sesuai dengan kebutuhan. Data berisi laporan harga yang dilaporkan 61 hari sebelumnya yang masing-masing laporan harga terdiri atas `laporanHarga_id`, `datePost`, `user_id`, `komoditas_id`, `harga`, `latitude`, `longitude`, `alamat`, `namaKomoditas`, `satuan`, dan `nama` pengirim laporan harga.

```

1 {
2   "status": 200,
3   "message": "sukses mendapat laporan harga 61 hari sebelumnya",
4   "data": [
5     {
6       "laporanHarga_id": 49,
7       "datePost": "2017-06-08",
8       "user_id": 67,
9       "komoditas_id": 13,
10      "harga": 20000,
11      "latitude": -6.5363512,
12      "longitude": 106.7313614,
13      "alamat": "Jl. Cimulang No.28, Bantarjaya, Ranca Bungur, Bogor, Jawa Barat 16310, Indonesia",
14      "namaKomoditas": "Bawang",
15      "satuan": "Kg",
16      "nama": "Irfan Rafii Musyafa "
17    },
18    {
19      "laporanHarga_id": 46,
20      "datePost": "2017-06-08",
21      "user_id": 67,
22      "komoditas_id": 11,
23      "harga": 30000,
24      "latitude": -6.5595708,
25      "longitude": 106.728302,
26      "alamat": "Jl. Kamper, Babakan, Dramaga, Bogor, Jawa Barat 16680, Indonesia",
27      "namaKomoditas": "Kedelai",
28      "satuan": "Kg",
29      "nama": "Irfan Rafii Musyafa "
30    },
31    {
32      "laporanHarga_id": 43,
33      "datePost": "2017-06-08",
34      "user_id": 67,

```

Gambar 15 Pengujian fungsi melihat laporan harga 61 hari sebelumnya

Tabel 4 Hasil pengujian iterasi kedua

Class	Method	Fungsi	URL	Status
Laporan harga	POST	Mendukung operasi pasar	<code>operasiPasar/pendukung/add</code>	Berhasil

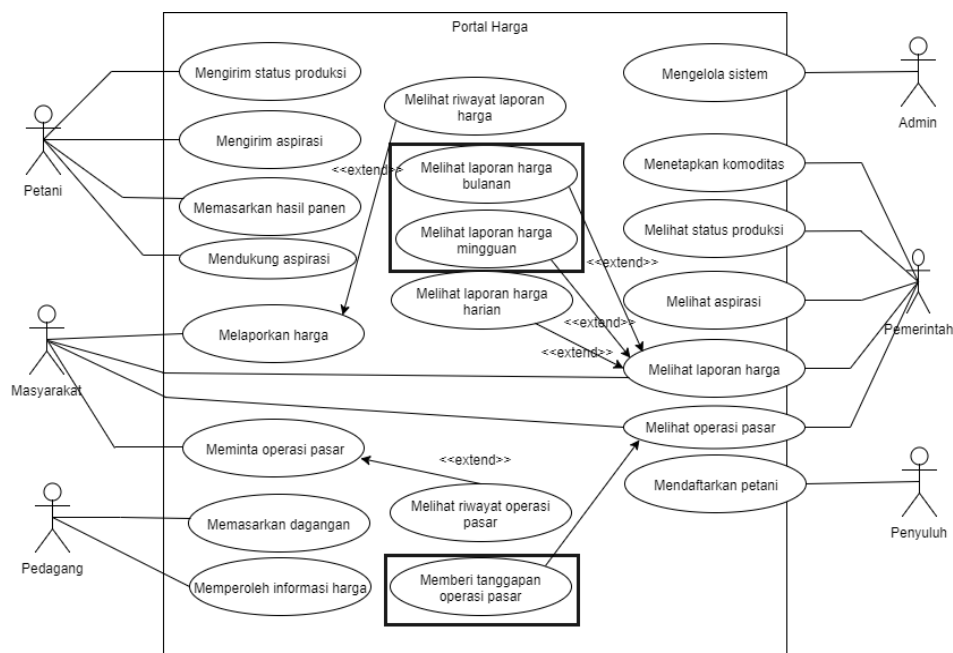
Tabel 4 Lanjutan

Class	Method	Fungsi	URL	Status
Operasi pasar	POST	Membatalkan dukung operasi pasar	operasiPasar/pendukung/del ete	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
	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.

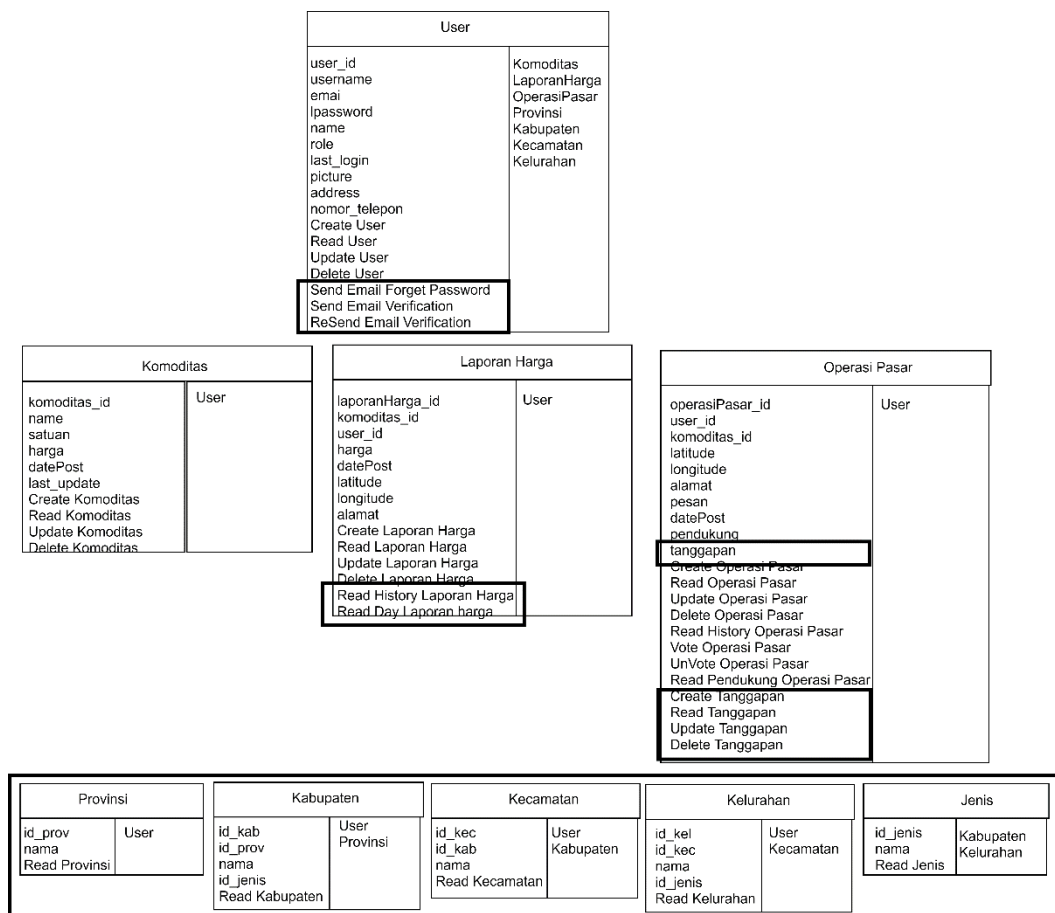


Gambar 16 Use case diagram iterasi ketiga

Modul *email* dikembangkan untuk verifikasi *account*. Verifikasi *account* dikirim melalui *email* dan jika tidak diverifikasi maka tidak bisa *login*. Pengguna yang melakukan *register* secara otomatis akan mendapatkan *email* yang berisi *link* untuk verifikasi *account*. Apabila selama satu jam pengguna tidak memverifikasi *account* maka *link* akan kedaluwarsa. Pengguna dapat meminta kembali *email* yang berisi *link* verifikasi dengan memasukkan *username*. Detil *use case diagram* iterasi ketiga dapat dilihat pada Gambar 16. Fungsi tiap modul yang dikembangkan di iterasi ketiga pada *use case diagram* diberi kotak hitam.

Desain (Design)

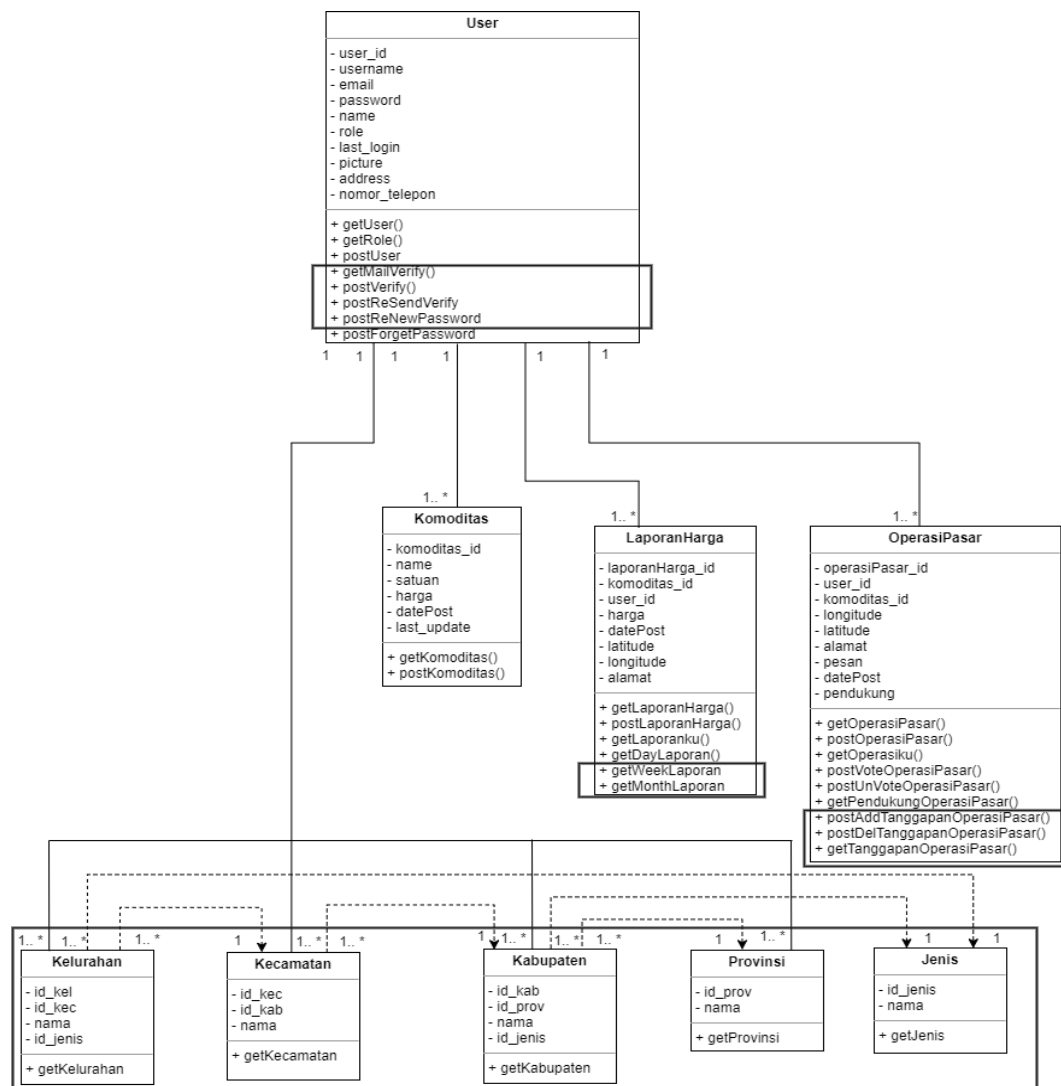
CRC *cards* dirancang menjadi *class diagram* sesuai modul yang dikembangkan pada tahap perencanaan yang dapat dilihat pada Gambar 17. *Class diagram* iterasi ketiga dapat dilihat pada Gambar 18. *Sequence diagram* dirancang sesuai dengan hubungan antar *object-object* yang dikembangkan pada *class diagram*. Detil *sequence diagram* iterasi ketiga dapat dilihat pada Lampiran 10. *Collaboration diagram* iterasi ketiga dirancang sesuai dengan *object-object* yang ada pada *sequence diagram*. Detil *Collaboration diagram* iterasi ketiga dapat dilihat pada Lampiran 11.



Gambar 17 CRC cards iterasi ketiga

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.

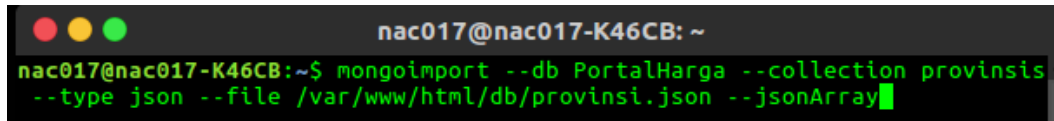


Gambar 18 *Class diagram* iterasi ketiga

Pengodean (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 dimasukkan 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 19. *Query* tersebut memasukan data provinsi dalam format JSON yang berada dalam folder `var/www/html/db` ke dalam *collections* *provinsis* di basis data PortalHarga.



```

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

```

Gambar 19 Code memasukan data format JSON ke basis data mongoDB

Pair programming dilakukan antara fungsi `getMailVerify` dengan fungsi `register` yang dapat dilihat pada Gambar 20. *Email* yang dikirim ke pengguna berisi *link* yang diintegrasikan dengan fungsi `postVerify`. Fungsi `postMailVerify` untuk mengubah *field* `isValidate` pada *collections* *user* agar bernilai *true* yang memiliki nilai *default* *false*.

```

//when user is created, sending email verification
mail.getMailVerify(req,res,user.isValiddate,user.email,user.username,user.name,user.user_id);
res.json({"status":"200","message": "Create User Success",data:user,token:token});

```

Gambar 20 Integrasi fungsi `getMailVerify` dengan fungsi *register*

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*. Pengujian pada iterasi ketiga dilakukan sesuai dengan *acceptance test criteria* iterasi ketiga yang dapat dilihat pada Lampiran 12. Detil pengujian iterasi ketiga dapat dilihat pada Tabel 5.

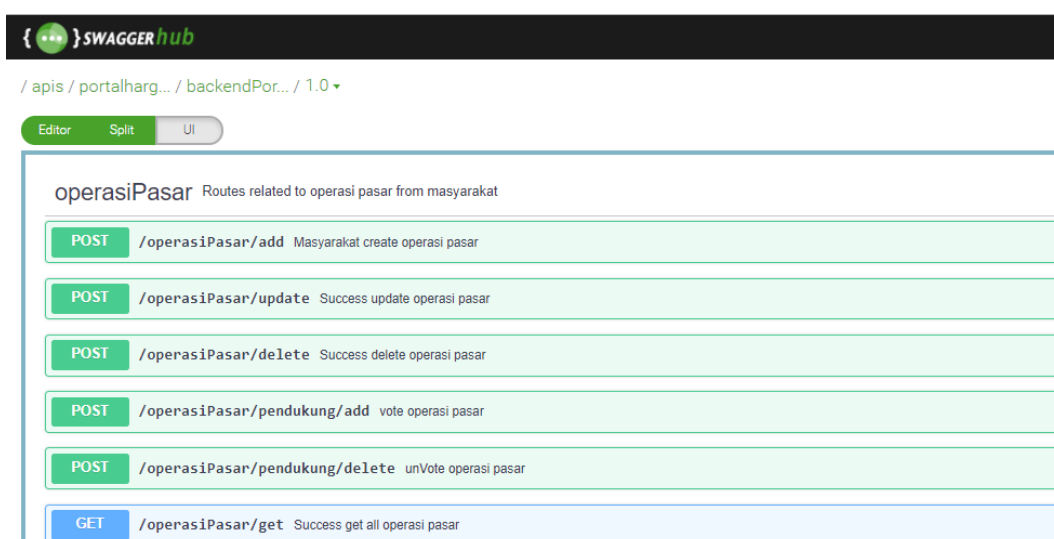
Salah satu pengujian pada modul lokasi, yaitu pengujian fungsi melihat provinsi di Indonesia. Pengujian fungsi melihat provinsi di Indonesia dengan mengakses URL `https://ph.yippytech.com:5000/lokasi/provinsi` menggunakan *method* GET. Data berisi informasi provinsi yang masing-masing terdiri dari `id_prov` dan nama provinsi. API yang sudah dikembangkan didokumentasikan di `https://app.swaggerhub.com/apis/portaltharga/backendPortalHarga/1.0`. Potongan dokumentasi API yang dapat dilihat pada Gambar 21.

Tabel 5 Hasil pengujian iterasi ketiga

<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

Tabel 5 Lanjutan

<i>Class</i>	<i>Method</i>	Fungsi	URL	Status
Mail	GET	Mengirim email verifikasi	/user/email/getMailVerify	Berhasil
	POST	Mengirim ulang email verifikasi	/user/email/validate/resent	Berhasil
Mail	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
Operasi pasar	POST	Memberi tanggapan operasi pasar	operasiPasar/tanggapan/add	Berhasil
	POST	Menghapus tanggapan operasi pasar	operasiPasar/tanggapan/delete	Berhasil
	GET	Melihat siapa saja yang memberi tanggapan	operasiPasar/get/:operasiPasar	Berhasil



Gambar 21 Dokumentasi API di Swaggerhub

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

- Ahmad. 2015. Perancangan aplikasi komoditas pertanian berbasis android. *Jurnal CSRID*. 7(3): 190–200.
- [APJII] Asosiasi Penyelenggara Jasa Internet Indonesia. 2016. Profil pengguna internet Indonesia 2016 [Internet]. [diunduh 2017 Aug 7]. Tersedia pada: <https://apjii.or.id/downfile/file/BULETINAPJIIEDISI02Oktober2016.pdf>.
- 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.
- [JSON] JavaScript Object Notation. 2002. Introduction JSON [internet]. [diunduh 2017 Jul 27]. Tersedia pada: <http://json.org/json-id.html>.
- [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.
- 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.

- Romadhan M. 2009. Pola hubungan tengkulak dan petani [skripsi]. Medan (ID): Universitas Sumatera Utara.
- 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.

Lampiran 1 Deskripsi *user story*

a Pemerintah

<i>Actor</i>	<i>Task</i>	<i>User story</i>
Pemerintah	Menetapkan komoditas	Pemerintah menentukan komoditas apa saja yang boleh masuk ke dalam sistem beserta harga dan satuannya
	Melihat status produksi	Status produksi yang dikirimkan oleh petani dapat dilihat oleh pemerintah, sehingga produksi yang dihasilkan oleh petani dapat dipantau oleh pemerintah
	Melihat operasi pasar	Pemerintah dapat melihat operasi pasar yang diminta oleh masyarakat sehingga pemerintah bisa mengatasi harga suatu komoditas di suatu wilayah tertentu
	Melihat laporan harga	Laporan harga yang dikirim oleh masyarakat dapat dilihat oleh pemerintah untuk menentukan kebijakan harga selanjutnya
	Melihat aspirasi	Aspirasi petani yang dikirim dapat dilihat oleh pemerintah sehingga pemerintah segera menangani kebutuhan petani yang ada di lapangan

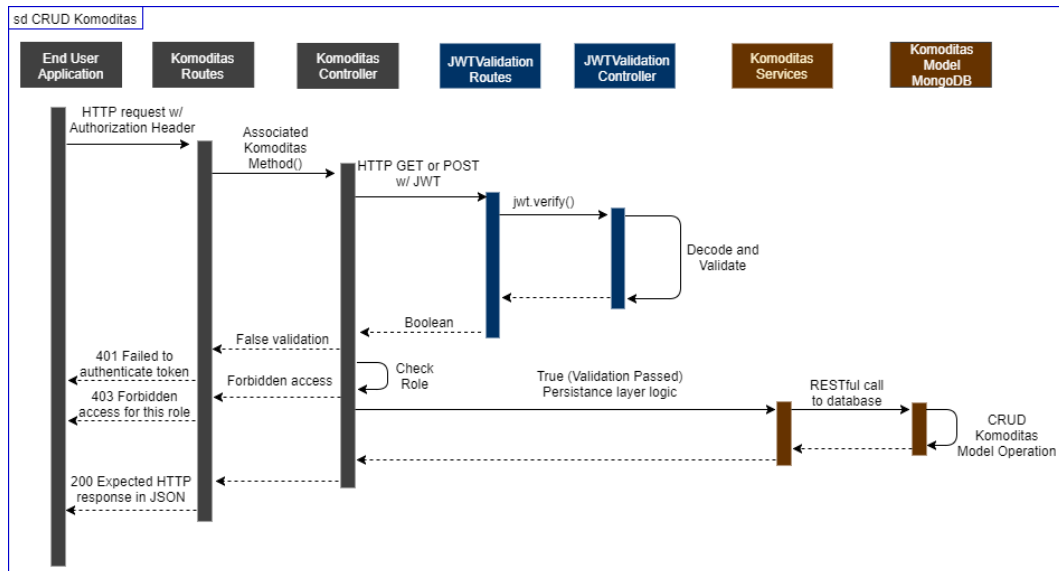
b Pedagang

<i>Actor</i>	<i>Task</i>	<i>User story</i>
Pedagang	Melihat laporan harga	Pedagang dapat melihat laporan harga yang dikirimkan oleh masyarakat sehingga pedagang bisa menentukan harga jual untuk suatu komoditas
	Memasarkan dagangan	Pedagang bisa memasarkan komoditas yang mereka dagangkan

Lampiran 2 *Sequence diagram* iterasi pertama

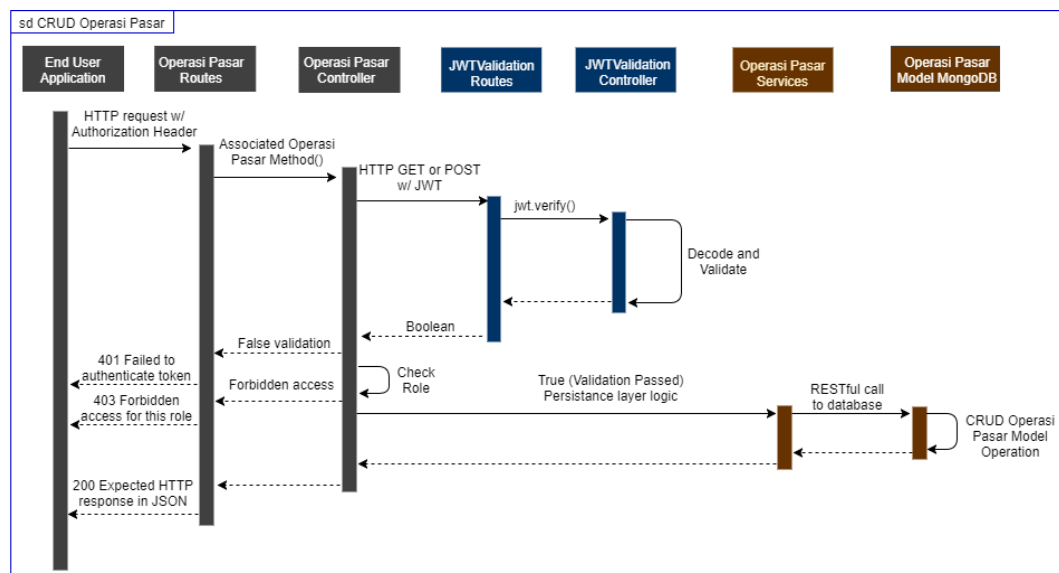
Komoditas

a CRUD Komoditas



Operasi pasar

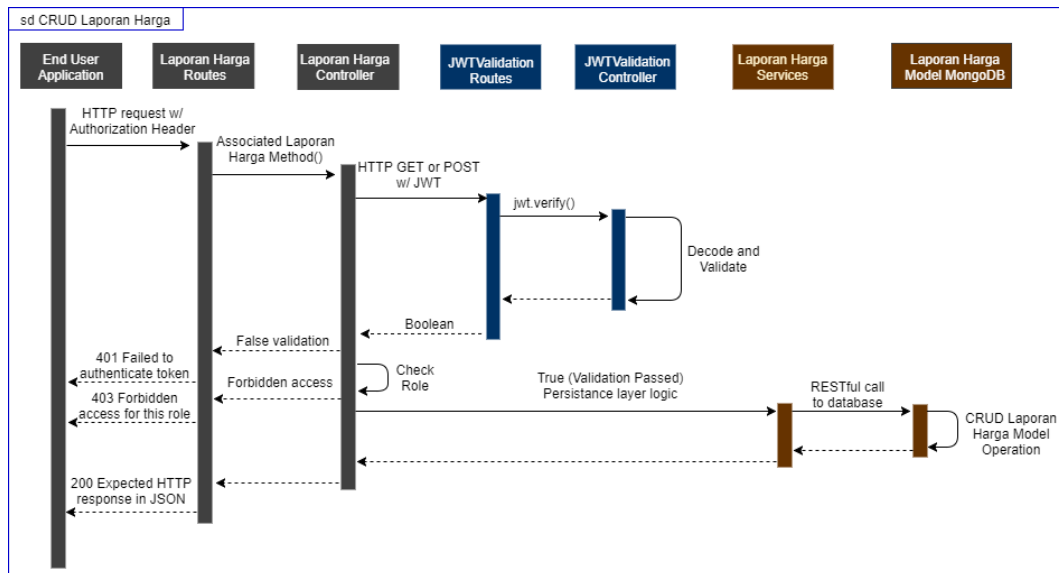
b CRUD Operasi pasar



Lampiran 2 *Sequence diagram* iterasi pertama (lanjutan)

Laporan harga

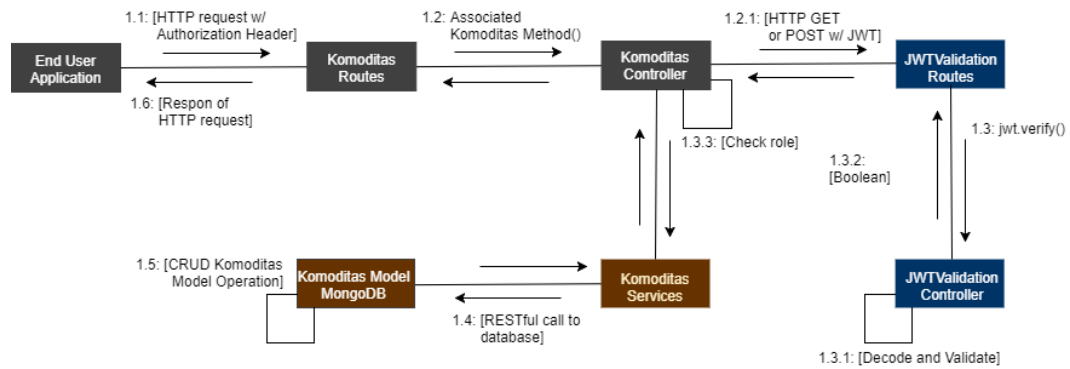
c CRUD Laporan harga



Lampiran 3 Collaboration diagram iterasi pertama

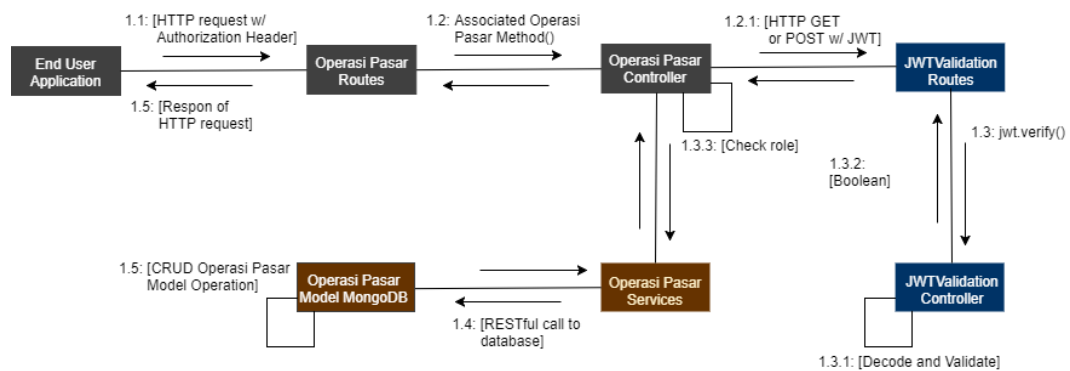
Komoditas

a CRUD Komoditas



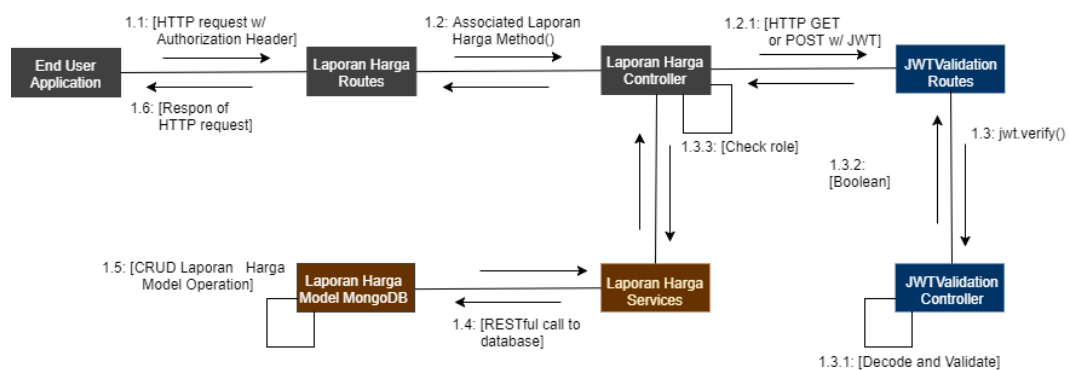
Operasi pasar

b CRUD Operasi pasar



Laporan harga

c CRUD Laporan harga



Lampiran 4 Daftar *collections* iterasi pertamaa *Collections* laporan harga

```

1  var mongoose=require('mongoose'),
2      Schema=mongoose.Schema,
3      autoIncrement=require('mongoose-auto-increment');
4
5  var connection = mongoose.createConnection("mongodb://localhost/PortalHarga");
6
7  autoIncrement.initialize(connection);
8
9  var laporanHargaModel = new Schema({
10     komoditas_id:Number,
11     user_id:Number,
12     harga:Number,
13     datePost:Number,
14     //lokasi:[{
15     latitude:Number,
16     longitude:Number,
17     alamat:String
18     //}]
19 });
20
21 laporanHargaModel.plugin(autoIncrement.plugin,{model:'laporanHarga',
22 field:'laporanHarga_id',startAt:1});
23
24 module.exports = mongoose.model("laporanHarga",laporanHargaModel);

```

b *Collections* operasi pasar

```

1  var mongoose=require('mongoose'),
2      Schema=mongoose.Schema,
3      autoIncrement=require('mongoose-auto-increment');
4
5  var connection = mongoose.createConnection("mongodb://localhost/PortalHarga");
6
7  autoIncrement.initialize(connection);
8
9  var operasiPasarModel = new Schema({
10     user_id:Number,
11     komoditas_id:Number,
12     latitude:Number,
13     longitude:Number,
14     alamat:String,
15     pesan:String,
16     datePost:Number,
17     //array pendukungnya
18     pendukung:[{
19         user_id:Number
20     }],
21     tanggapan:[
22         {
23             user_id:String,
24             isi:String,
25             datePost:Number
26         }
27     ]
28 });
29
30 operasiPasarModel.plugin(autoIncrement.plugin,{model:'operasiPasar',field:'operasiPasar_id',startAt:1});
31
32 module.exports = mongoose.model("operasiPasar",operasiPasarModel);

```

Lampiran 5 Kamus data

Komoditas

Field name	Caption	Data Type	Key Type	Constraints	Reference Collections
komoditas_id		Autonumber	Primary Key		
name	Name komoditas	String		Not Null	
satuan	Satuan komoditas	String		Not Null	
harga	Harga komoditas	Number		Not Null	
datePost	Date input	Number			
last_update	Date update	Number			

Laporan harga

Field name	Caption	Data Type	Key Type	Constraints	Reference Collections
laporanHarga_id		Autonumber	Primary Key		
komoditas_id	Name komoditas	Number	Foreign key	Not Null	komoditas
user_id	Satuan komoditas	Number	Foreign key	Not Null	users
harga	Harga komoditas	Number		Not Null	
datePost	Date input	Number			
latitude	Coordinate latitude	Number		Not Null	
longitude	Coordinate longitude	Number		Not Null	
alamat	Location details	String		Not Null	

Operasi pasar

Field name	Caption	Data Type	Key Type	Constraints	Reference Collections
operasiPasar_id		Autonumber	Primary Key		
user_id	Name komoditas	Number	Foreign key	Not Null	users

Lampiran 5 Kamus data (lanjutan)

Field name	Caption	Data Type	Key Type	Constraints	Reference Collections
komoditas_id	Satuan komoditas	Number	Foreign key	Not Null	komoditas
latitude	Coordinate latitude	Number		Not Null	
longitude	Coordinate longitude	Number		Not Null	
alamat	Location details	String		Not Null	
pesan	Message user	String		Not Null	
datePost	Date input	Number			
pendukung	Vote of users	Array		Not Null	
tanggapan	Response	Array		Not Null	

Lampiran 6 *Acceptance test criteria* iterasi pertama

a Komoditas

ID Pengujian	Deskripsi Pengujian	Prosedur Pengujian	Data masukan	Keluaran yang diharapkan (JSON)
Kd.1	Menambah komoditas	<ul style="list-style-type: none"> Memilih <i>method</i> POST Klik tombol <i>send</i> 	<ul style="list-style-type: none"> token nama komoditas satuan komoditas harga komoditas 	<ul style="list-style-type: none"> status <i>message</i> data komoditas token
Kd.2	Melihat semua komoditas	<ul style="list-style-type: none"> Memilih <i>method</i> GET Klik tombol <i>send</i> 	<ul style="list-style-type: none"> token 	<ul style="list-style-type: none"> status <i>message</i> data komoditas token
Kd.3	Melihat detil satu komoditas	<ul style="list-style-type: none"> Memilih <i>method</i> GET Klik tombol <i>send</i> 	<ul style="list-style-type: none"> token komoditas_id 	<ul style="list-style-type: none"> status <i>message</i> data komoditas token
Kd.4	Mengubah informasi komoditas	<ul style="list-style-type: none"> Memilih <i>method</i> POST Klik tombol <i>send</i> 	<ul style="list-style-type: none"> token komoditas_id nama komoditas satuan komoditas harga komoditas 	<ul style="list-style-type: none"> status <i>message</i> data komoditas token
Kd.5	Menghapus komoditas	<ul style="list-style-type: none"> Memilih <i>method</i> POST Klik tombol <i>send</i> 	<ul style="list-style-type: none"> token komoditas_id 	<ul style="list-style-type: none"> status <i>message</i> data komoditas token

Lampiran 6 *Acceptance test criteria* iterasi pertama (lanjutan)

b Laporan harga

ID Pengujian	Deskripsi Pengujian	Prosedur Pengujian	Data Masukan	Keluaran yang diharapkan (JSON)
Lp.1	Menambah laporan harga	<ul style="list-style-type: none"> Memilih <i>method</i> POST Klik tombol <i>send</i> 	<ul style="list-style-type: none"> token komoditas_id user_id harga latitude longitude alamat token 	<ul style="list-style-type: none"> status <i>message</i> data laporan harga token
Lp.2	Melihat semua laporan harga	<ul style="list-style-type: none"> Memilih <i>method</i> GET Klik tombol <i>send</i> 	<ul style="list-style-type: none"> token 	<ul style="list-style-type: none"> status <i>message</i> data laporan harga token
Lp.3	Melihat detail satu laporan harga	<ul style="list-style-type: none"> Memilih <i>method</i> GET Klik tombol <i>send</i> 	<ul style="list-style-type: none"> token laporanharga_id 	<ul style="list-style-type: none"> status <i>message</i> data laporan harga token
Lp.4	Mengubah informasi laporan harga	<ul style="list-style-type: none"> Memilih <i>method</i> POST Klik tombol <i>send</i> 	<ul style="list-style-type: none"> token laporanharga_id harga komoditas 	<ul style="list-style-type: none"> status <i>message</i> data laporan harga token
Lp.5	Menghapus laporan harga	<ul style="list-style-type: none"> Memilih <i>method</i> POST Klik tombol <i>send</i> 	<ul style="list-style-type: none"> token laporanharga_id 	<ul style="list-style-type: none"> status <i>message</i> data laporan harga token

Lampiran 6 *Acceptance test criteria* iterasi pertama (lanjutan)

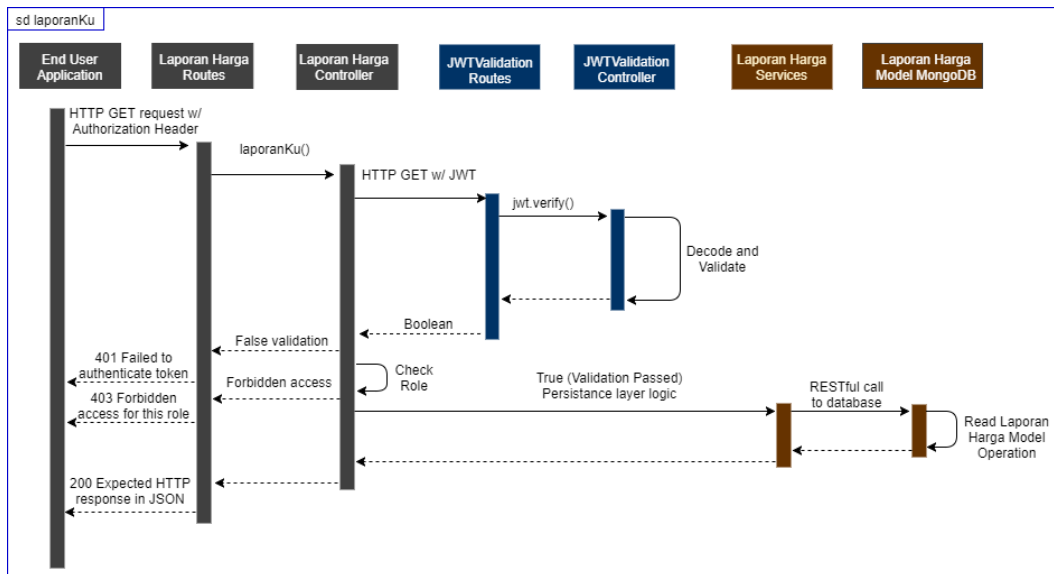
c Operasi pasar

ID Pengujian	Deskripsi Pengujian	Prosedur Pengujian	Data masukan	Keluaran yang diharapkan (JSON)
Op.1	Menambah operasi pasar	<ul style="list-style-type: none"> Memilih <i>method</i> POST Klik tombol <i>send</i> 	<ul style="list-style-type: none"> token user_id komoditas_id latitude longitude alamat pesan token 	<ul style="list-style-type: none"> status <i>message</i> data operasi pasar token
Op.2	Melihat semua operasi pasar	<ul style="list-style-type: none"> Memilih <i>method</i> GET Klik tombol <i>send</i> 	<ul style="list-style-type: none"> token 	<ul style="list-style-type: none"> status <i>message</i> data operasi pasar token
Op.3	Melihat detail satu operasi pasar	<ul style="list-style-type: none"> Memilih <i>method</i> GET Klik tombol <i>send</i> 	<ul style="list-style-type: none"> token operasiPasar_id 	<ul style="list-style-type: none"> status <i>message</i> data operasi pasar token
Op.4	Mengubah informasi operasi pasar	<ul style="list-style-type: none"> Memilih <i>method</i> POST Klik tombol <i>send</i> 	<ul style="list-style-type: none"> Token operasiPasar_id pesan 	<ul style="list-style-type: none"> status <i>message</i> data operasi pasar token
Op.5	Menghapus operasi pasar	<ul style="list-style-type: none"> Memilih <i>method</i> POST Klik tombol <i>send</i> 	<ul style="list-style-type: none"> token operasiPasar_id 	<ul style="list-style-type: none"> status <i>message</i> data operasi pasar token

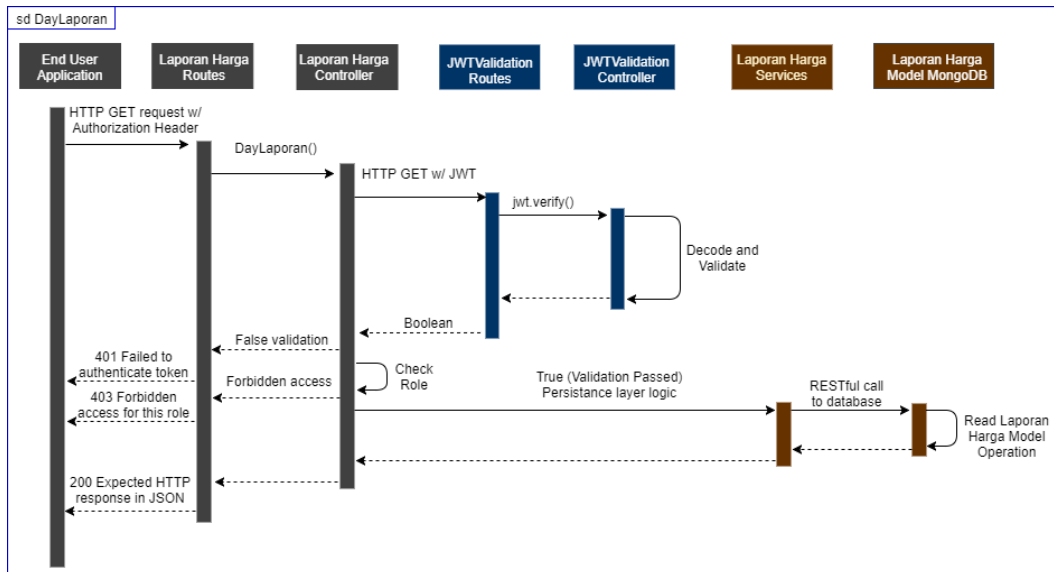
Lampiran 7 *Sequence diagram* iterasi kedua

Laporan Harga

a *History pengguna*



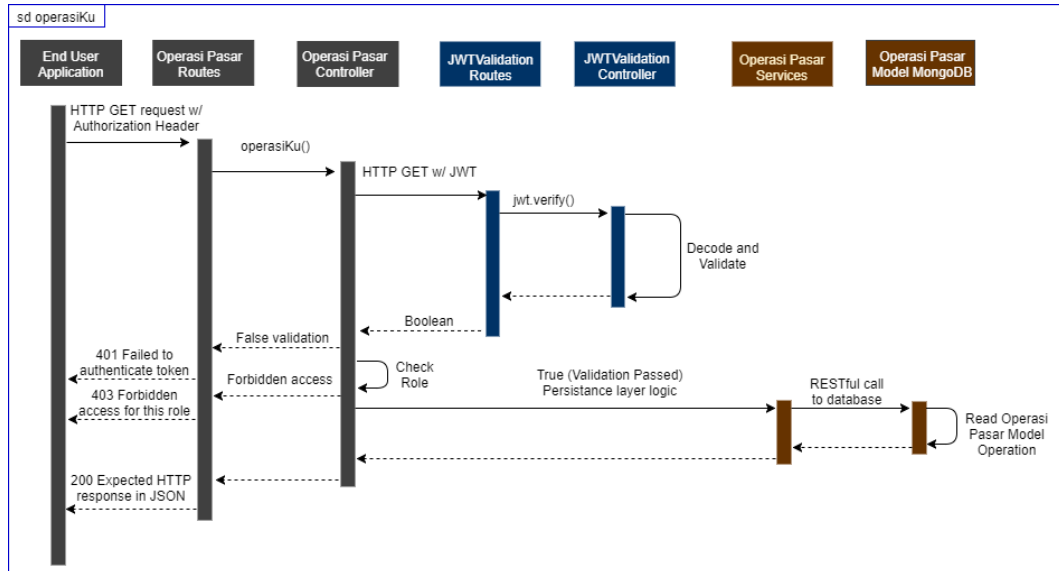
b *Day laporan*



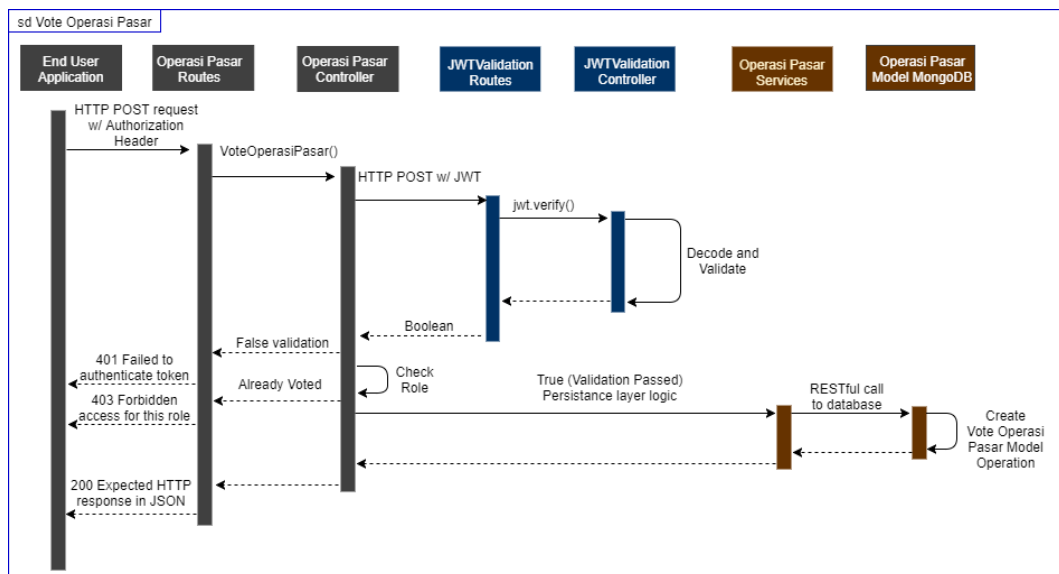
Lampiran 7 *Sequence diagram* iterasi kedua (lanjutan)

Operasi pasar

c *History pengguna*

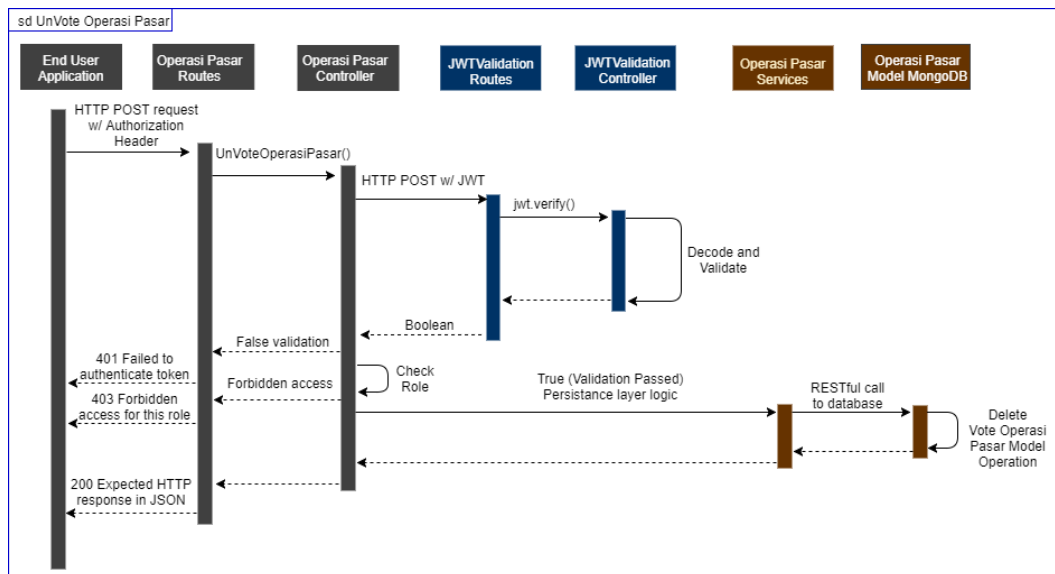


d *Vote operasi pasar*

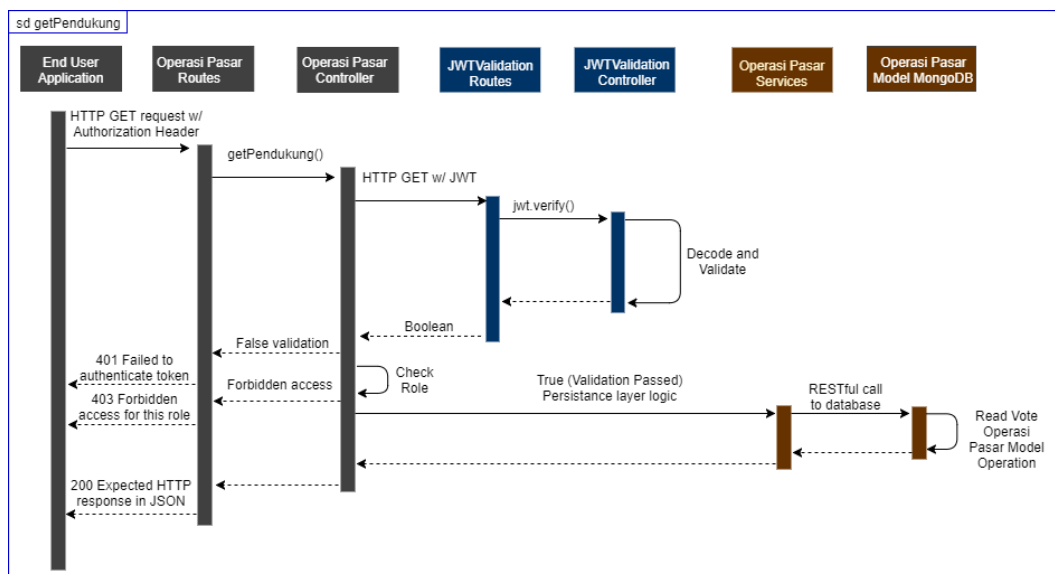


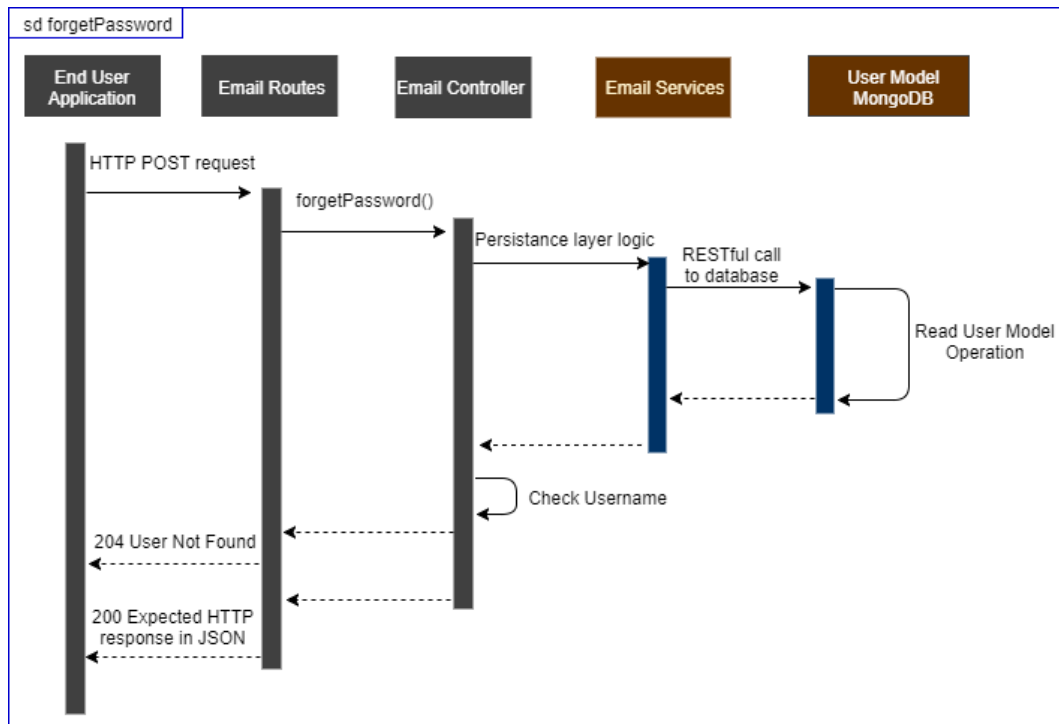
Lampiran 7 Sequence diagram iterasi kedua (lanjutan)

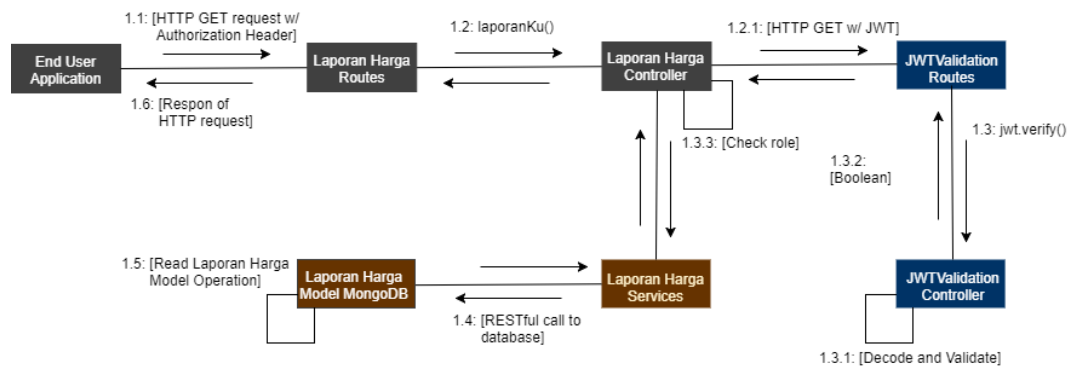
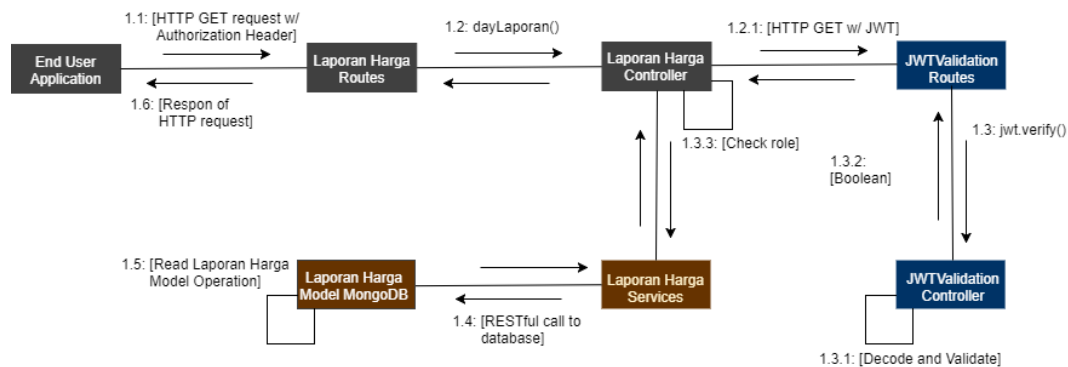
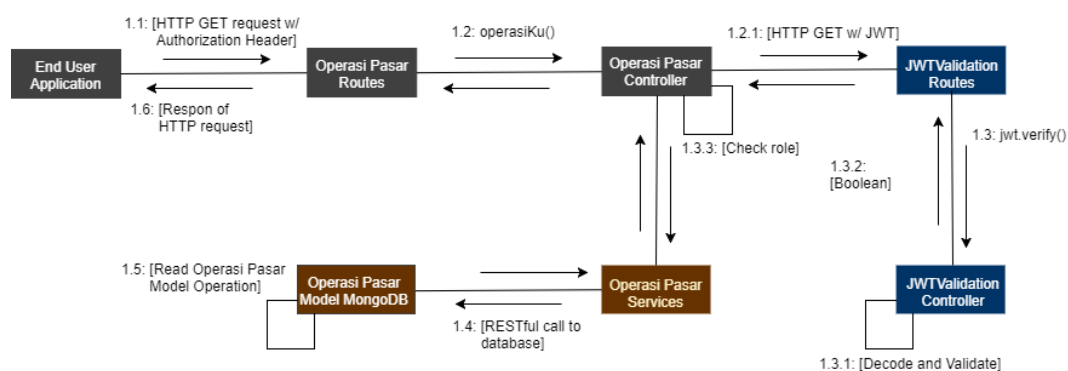
e *UnVote* operasi pasar



f *Get* pendukung operasi pasar

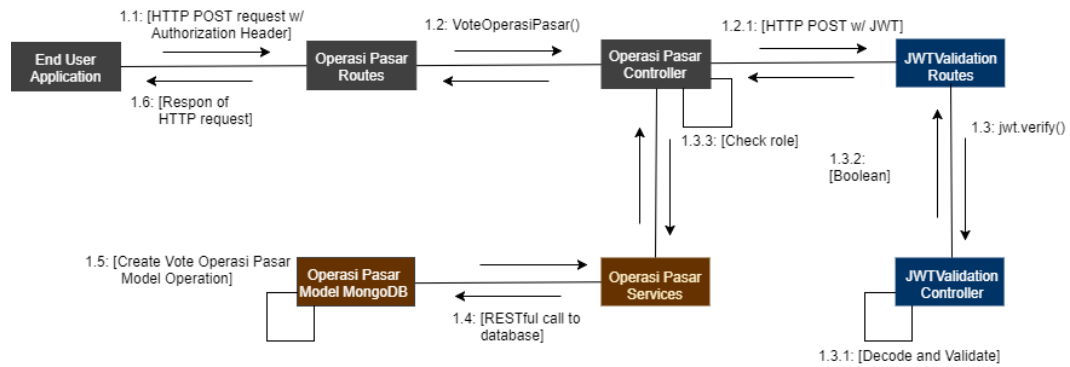


Lampiran 7 *Sequence diagram* iterasi kedua (lanjutan)**Email***g Forget password*

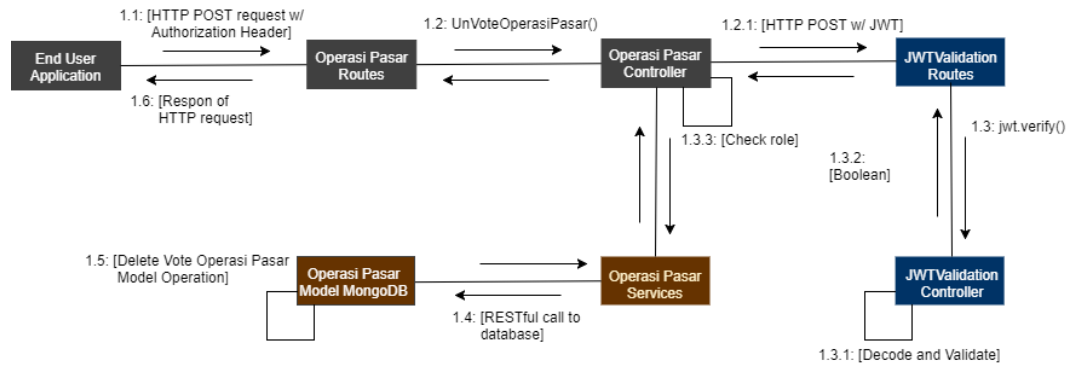
Lampiran 8 *Collaboration diagram* iterasi kedua**Laporan Harga****a** History pengguna**b** Day laporan**Operasi pasar****c** History pengguna

Lampiran 8 Collaboration diagram iterasi kedua (lanjutan)

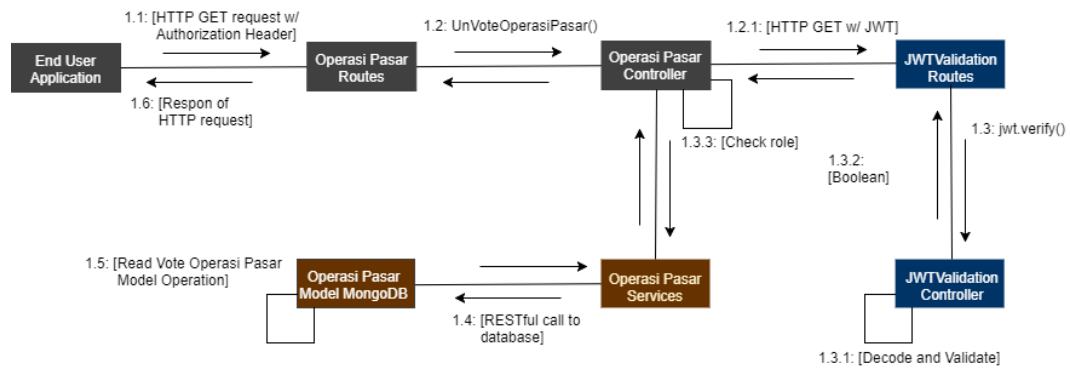
d *Vote operasi pasar*

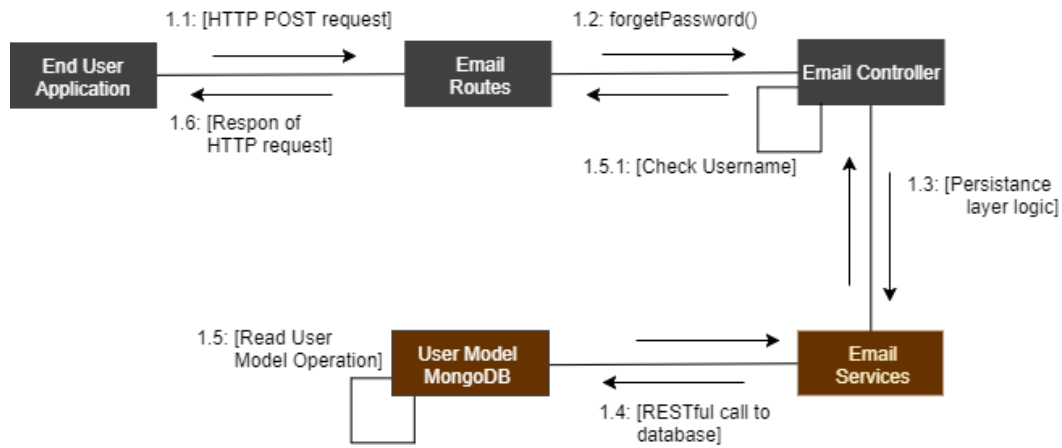


e *UnVote operasi pasar*



f *Get pendukung operasi pasar*



Lampiran 8 *Collaboration diagram* iterasi kedua (lanjutan)**Email***g Forget password*

Lampiran 9 *Acceptance test criteria* iterasi kedua

a Laporan harga

ID Pengujian	Deskripsi Pengujian	Prosedur Pengujian	Data masukan	Keluaran yang diharapkan (JSON)
Lp.6	Melihat <i>history</i> laporan harga seorang <i>user</i>	<ul style="list-style-type: none"> Memilih <i>method</i> GET Klik tombol <i>send</i> 	<ul style="list-style-type: none"> token <i>user_id</i> 	<ul style="list-style-type: none"> status <i>message</i> data laporan harga token
Lp.7	Melihat laporan harga pada suatu hari tertentu	<ul style="list-style-type: none"> Memilih <i>method</i> GET Klik tombol <i>send</i> 	<ul style="list-style-type: none"> token angka berapa hari dari sekarang 	<ul style="list-style-type: none"> status <i>message</i> data laporan harga token

b Operasi pasar

ID Pengujian	Deskripsi Pengujian	Prosedur Pengujian	Data masukan	Keluaran yang diharapkan (JSON)
Op.6	Melihat <i>history</i> operasi pasar seorang <i>user</i>	<ul style="list-style-type: none"> Memilih <i>method</i> GET Klik tombol <i>send</i> 	<ul style="list-style-type: none"> token <i>user_id</i> 	<ul style="list-style-type: none"> status <i>message</i> data operasi pasar token
Op.7	Mendukung operasi pasar	<ul style="list-style-type: none"> Memilih <i>method</i> POST Klik tombol <i>send</i> 	<ul style="list-style-type: none"> token <i>user_id</i> operasiPasar_id 	<ul style="list-style-type: none"> status <i>message</i> data operasi pasar token
Op.8	Membatalkan dukungan operasi pasar	<ul style="list-style-type: none"> Memilih <i>method</i> POST Klik tombol <i>send</i> 	<ul style="list-style-type: none"> token <i>user_id</i> operasiPasar_id 	<ul style="list-style-type: none"> status <i>message</i> data operasi pasar token

Lampiran 9 *Acceptance test criteria* iterasi kedua (lanjutan)

ID Pengujian	Deskripsi Pengujian	Prosedur Pengujian	Data masukan	Keluaran yang diharapkan (JSON)
Op.9	Melihat pendukung operasi pasar	<ul style="list-style-type: none"> Memilih <i>method</i> GET Klik tombol <i>send</i> 	<ul style="list-style-type: none"> token operasiPasar_id 	<ul style="list-style-type: none"> status <i>message</i> data operasi pasar token

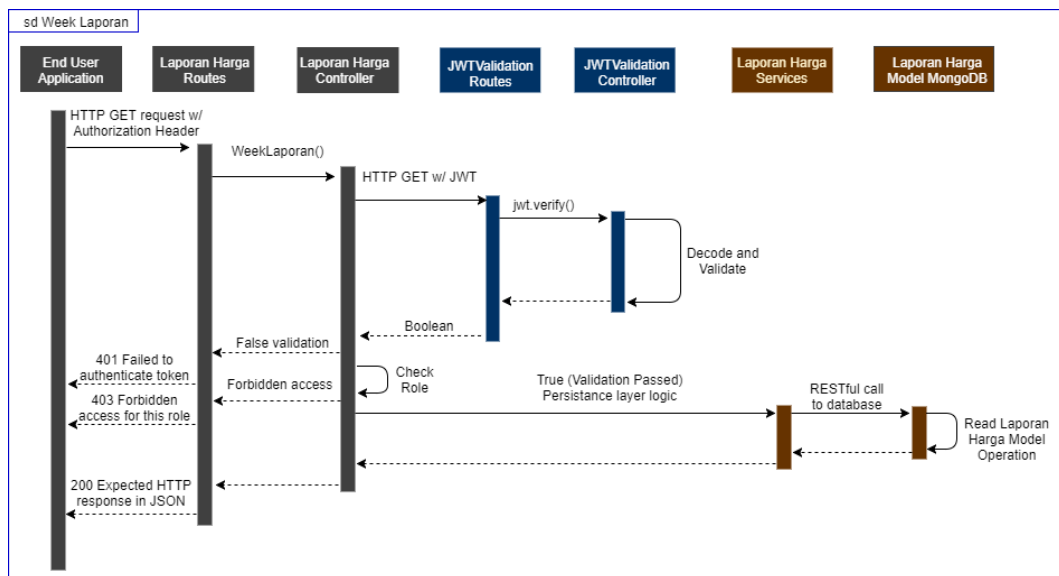
c Email

ID Pengujian	Deskripsi Pengujian	Prosedur Pengujian	Data masukan	Keluaran yang diharapkan (JSON)
Em.1	Mengirim password baru	<ul style="list-style-type: none"> Memilih <i>method</i> POST Klik tombol <i>send</i> 	<ul style="list-style-type: none"> username email 	<ul style="list-style-type: none"> status <i>message</i> data pengguna

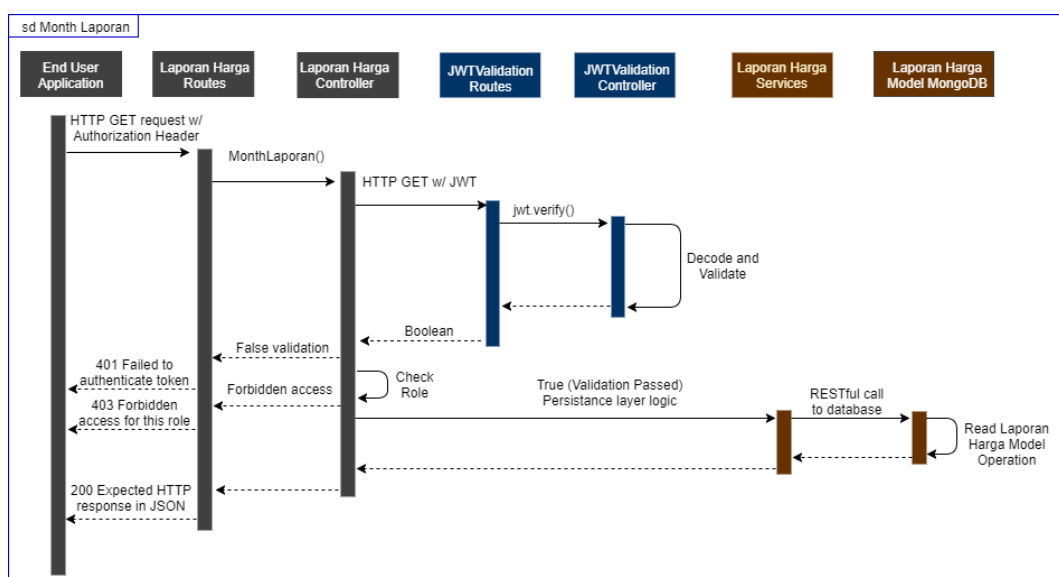
Lampiran 10 *Sequence diagram* iterasi ketiga

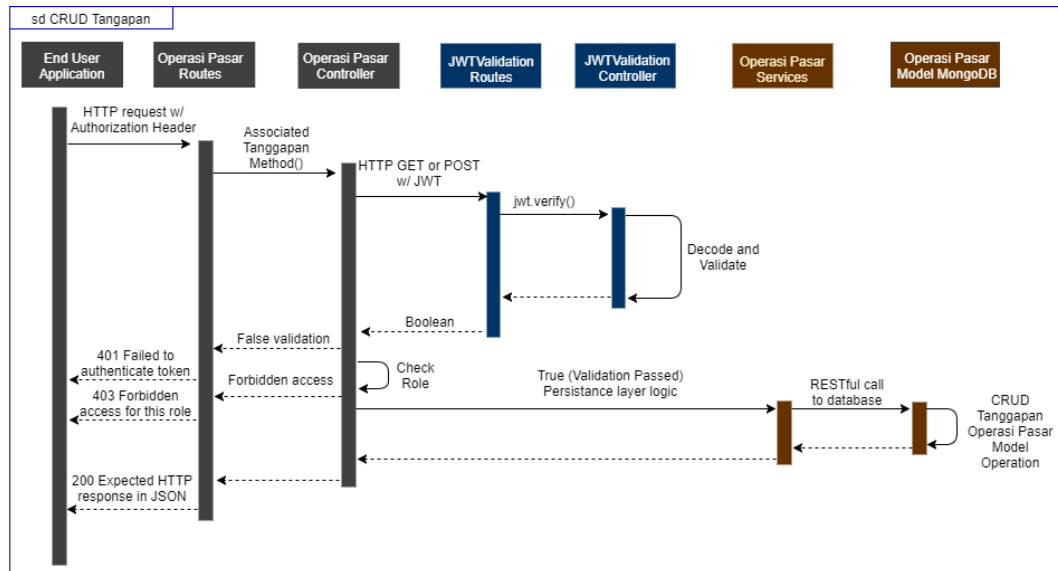
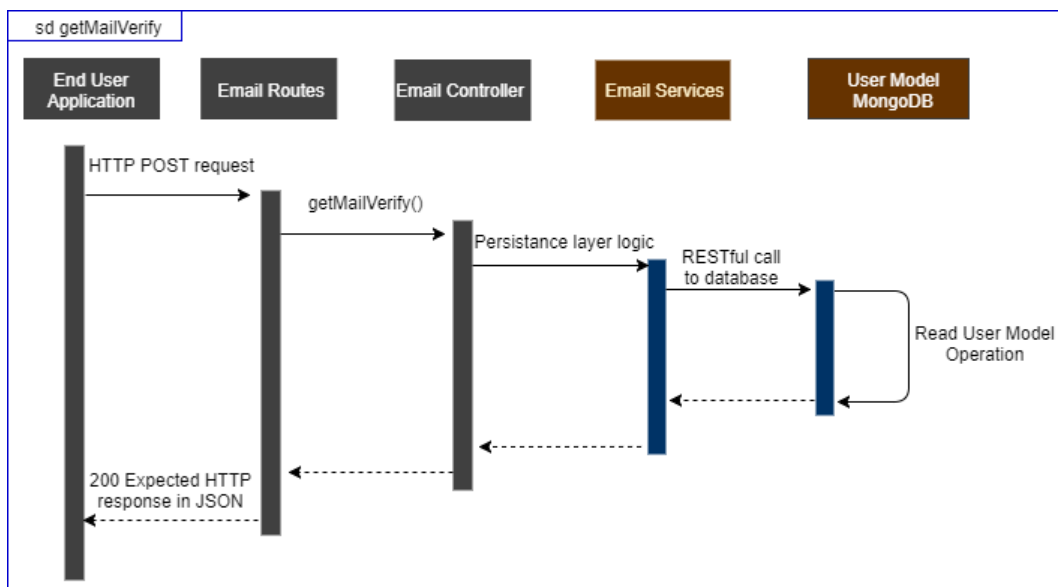
Laporan harga

a *Week* laporan harga



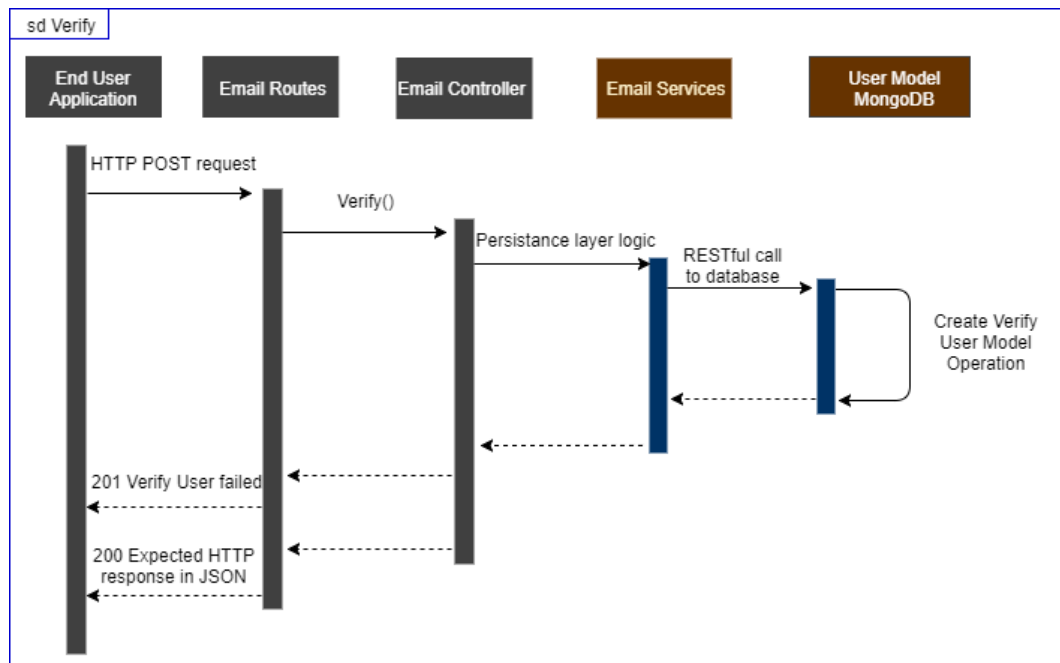
b *Month* laporan harga



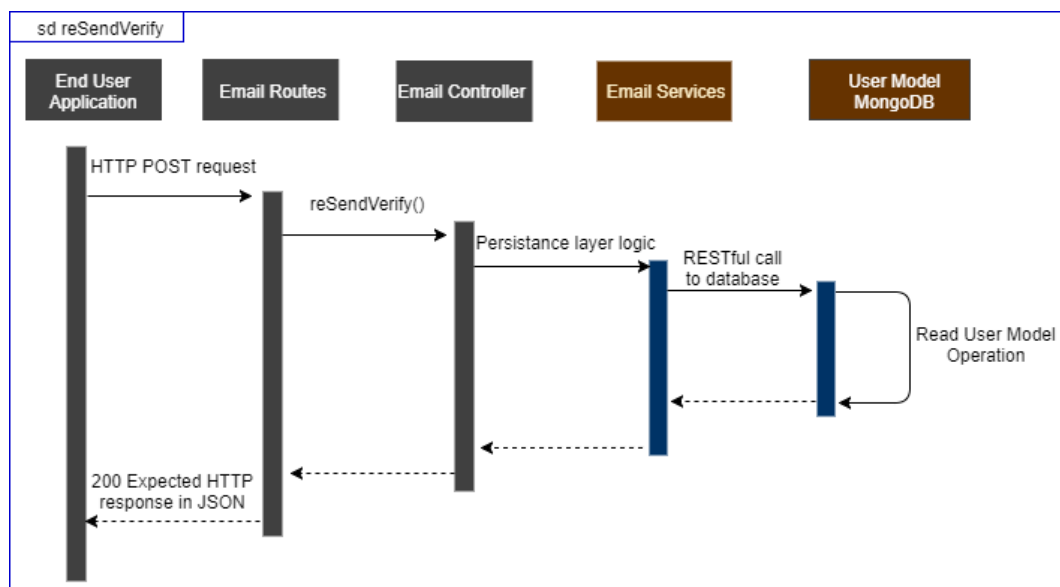
Lampiran 10 *Sequence diagram iterasi ketiga (lanjutan)***Operasi pasar****c** **CRUD tanggapan operasi pasar****Email****d** **getMailVerify**

Lampiran 10 *Sequence diagram* iterasi ketiga (lanjutan)

e Verify

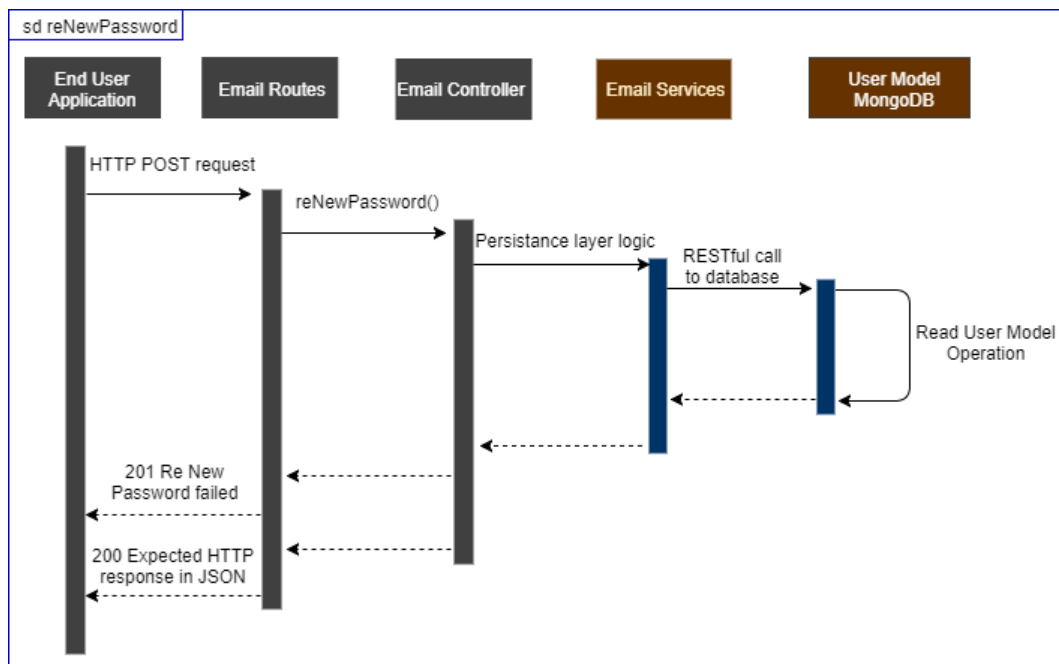


f reSendMailVerify



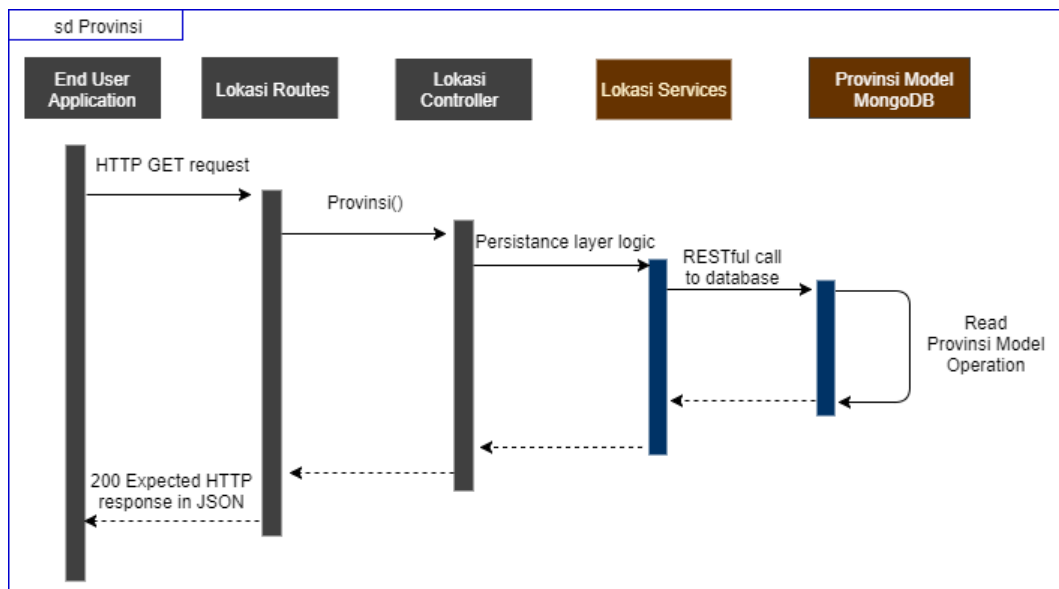
Lampiran 10 *Sequence diagram* iterasi ketiga (lanjutan)

g reNewPassword



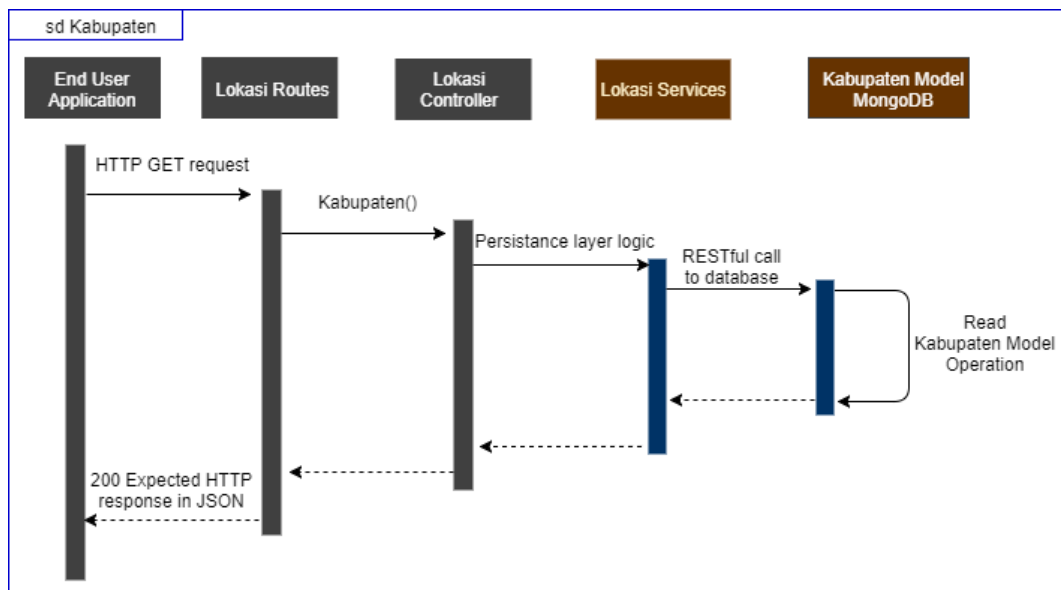
Lokasi

h Provinsi

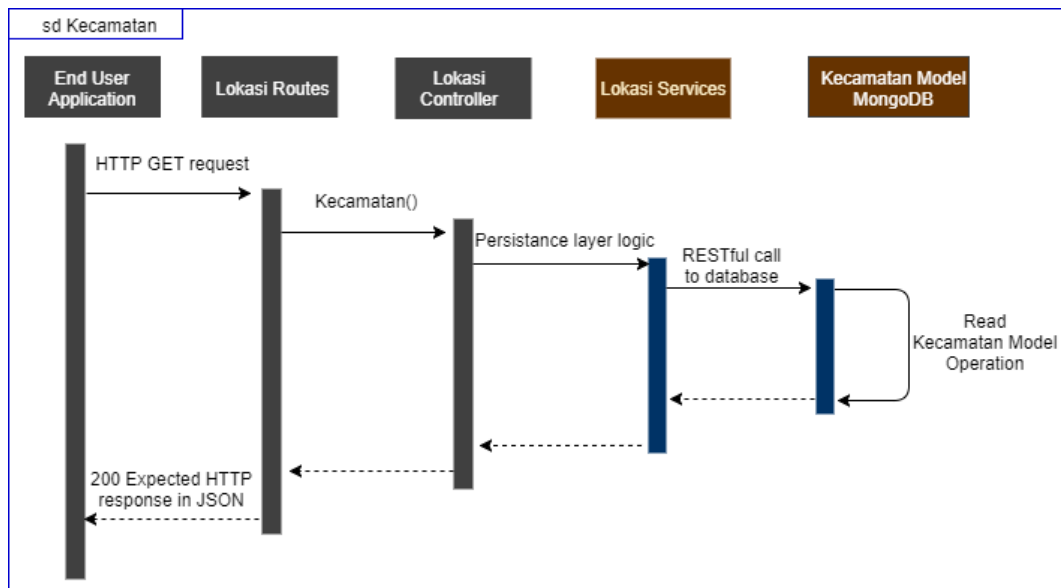


Lampiran 10 *Sequence diagram* iterasi ketiga (lanjutan)

i Kabupaten

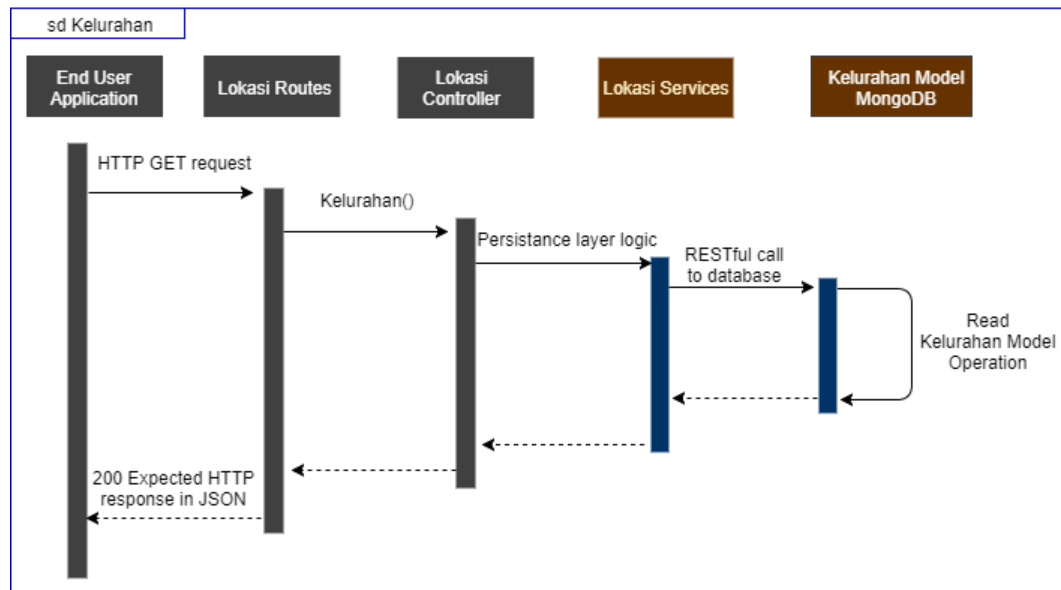


j Kecamatan



Lampiran 10 *Sequence diagram* iterasi ketiga (lanjutan)

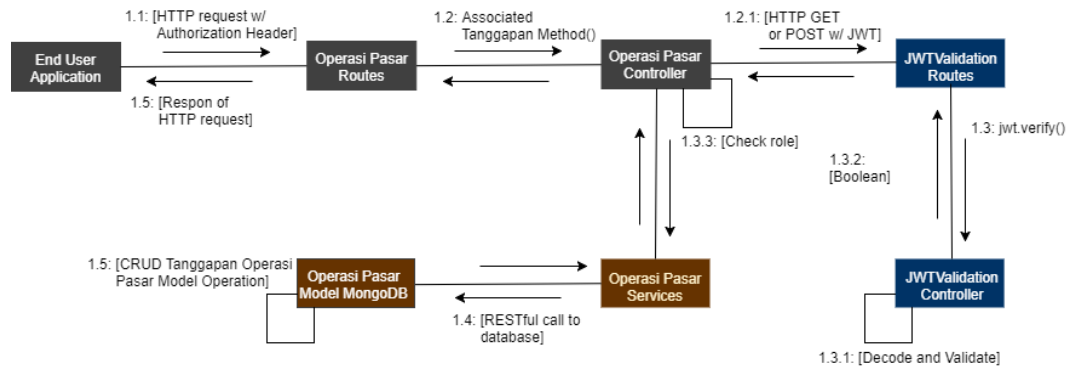
k Kelurahan



Lampiran 11 Collaboration diagram iterasi ketiga

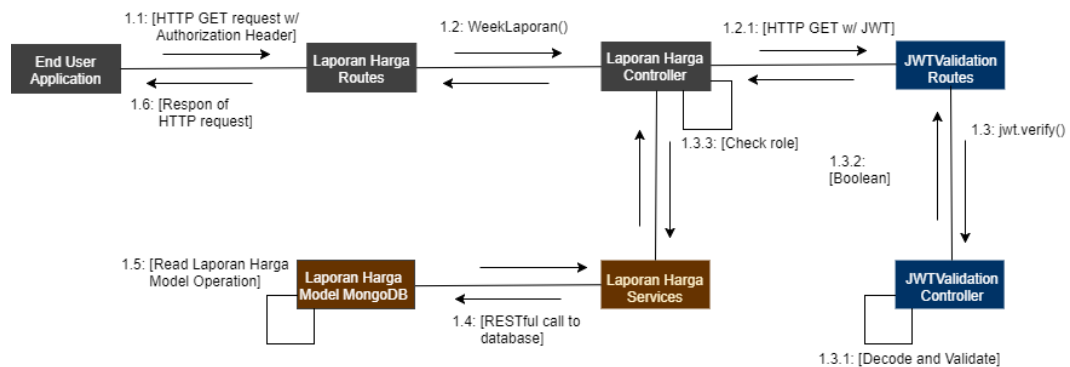
Operasi pasar

a CRUD tanggapan

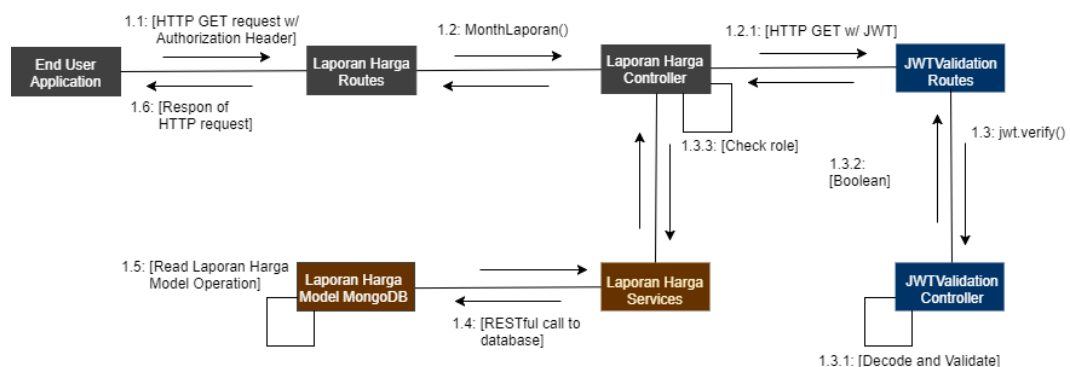


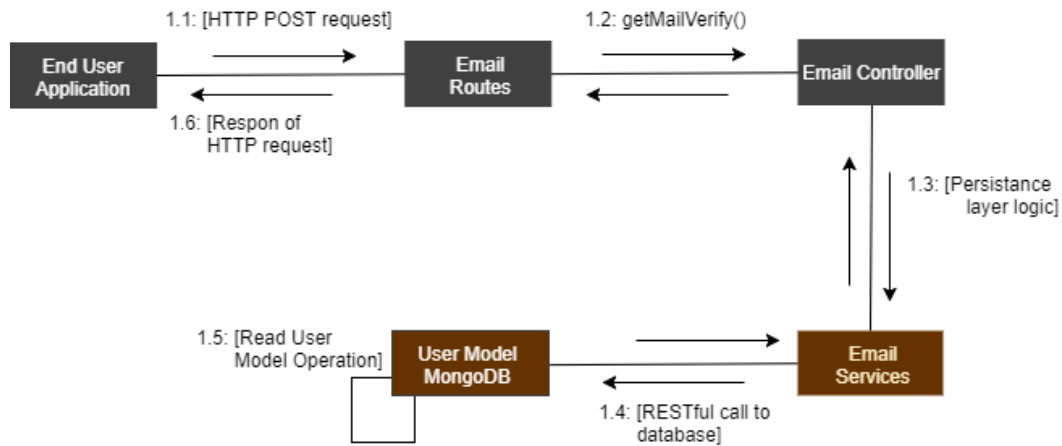
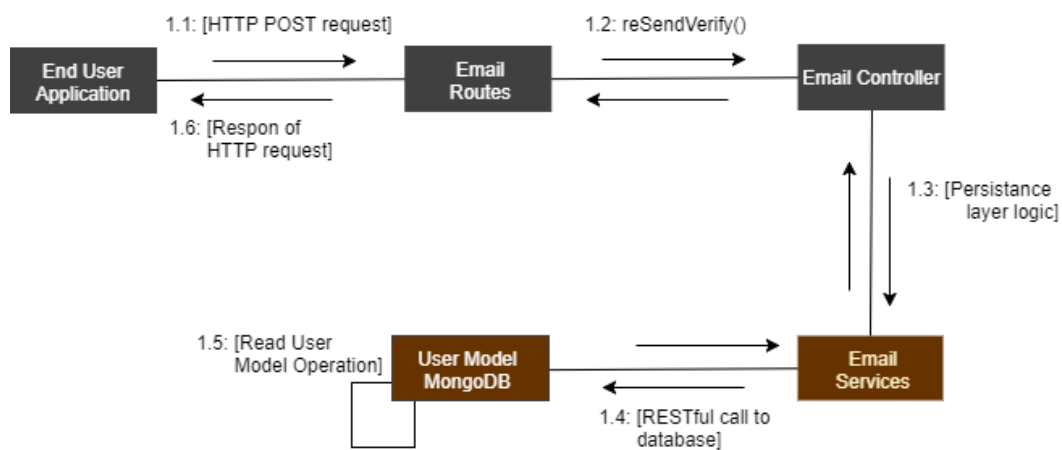
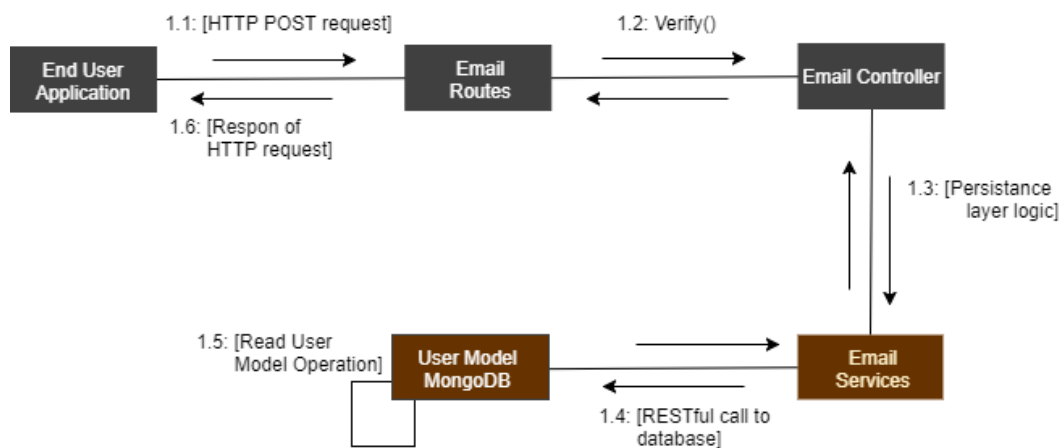
Laporan harga

b Week laporan



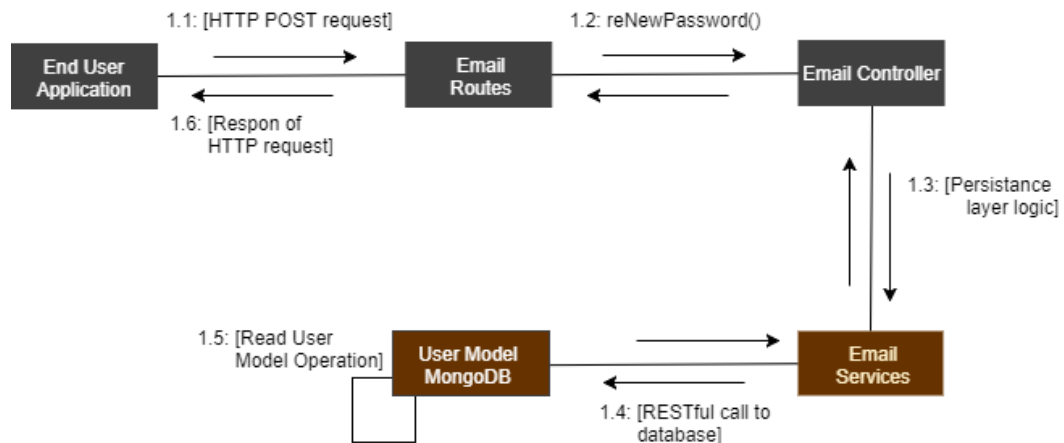
c Month laporan



Lampiran 11 *Collaboration diagram* iterasi ketiga (lanjutan)**Email**d `getMailVerify`e `reSendVerify`f `Verify`

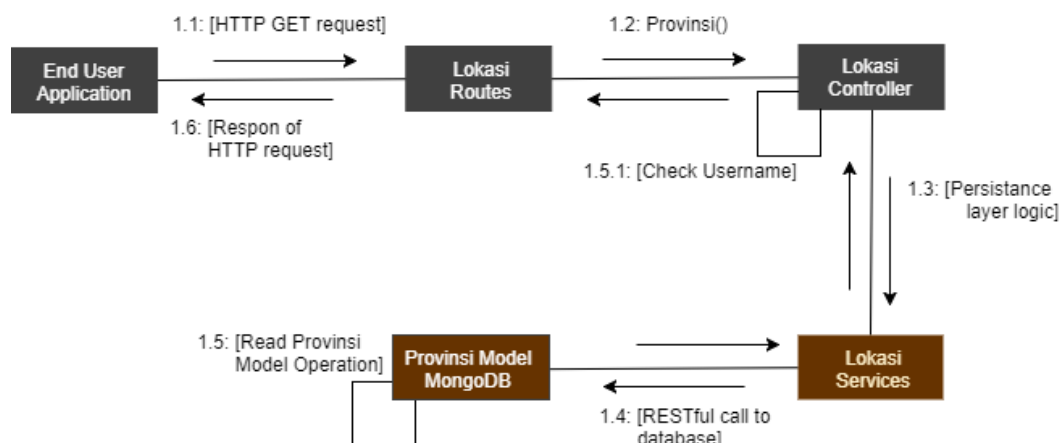
Lampiran 11 *Collaboration diagram* iterasi ketiga (lanjutan)

g reNewPassword

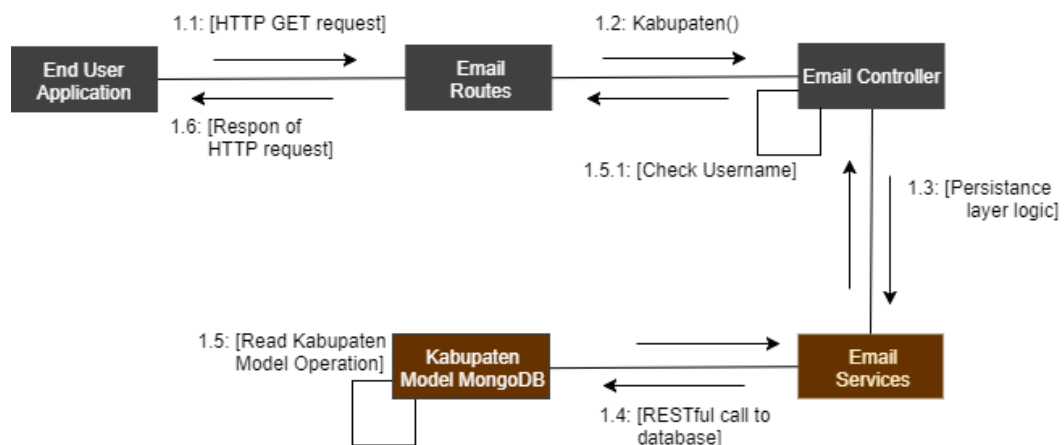


Lokasi

h Provinsi

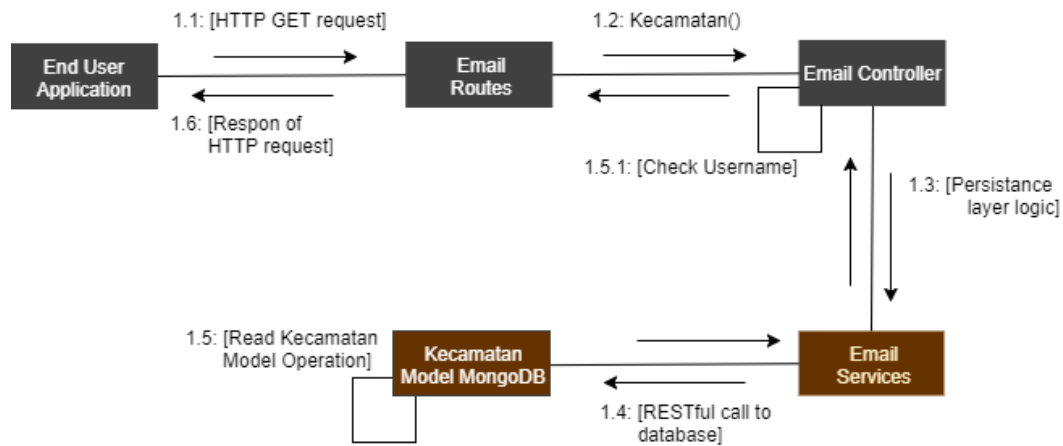


i Kabupaten

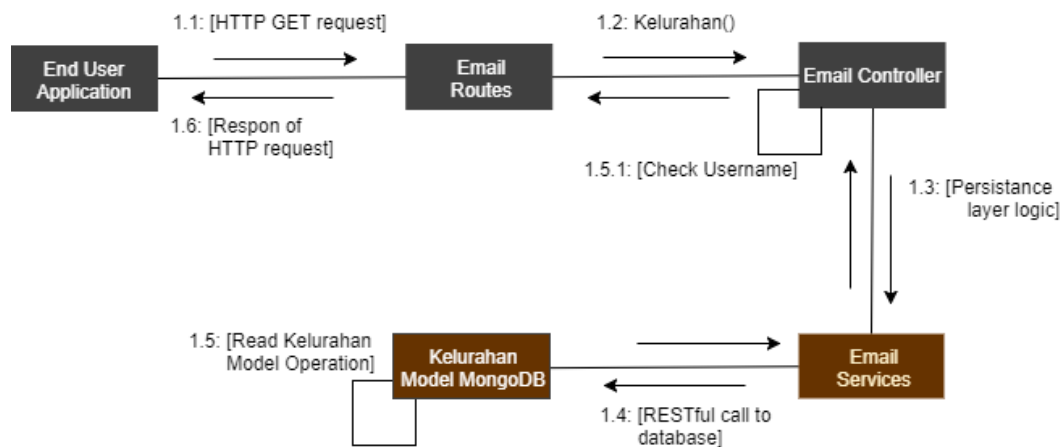


Lampiran 11 *Collaboration diagram* iterasi ketiga (lanjutan)

j Kecamatan



k Kelurahan



Lampiran 12 *Acceptance test criteria* iterasi ketiga

a Laporan harga

ID Pengujian	Deskripsi Pengujian	Prosedur Pengujian	Data Masukan	Keluaran yang diharapkan (JSON)
Lp.8	Melihat laporan harga minggu ini	<ul style="list-style-type: none"> Memilih <i>method</i> GET Klik tombol <i>send</i> 	<ul style="list-style-type: none"> token 	<ul style="list-style-type: none"> status <i>message</i> data laporan harga token
Lp.9	Melihat laporan harga bulan ini	<ul style="list-style-type: none"> Memilih <i>method</i> GET Klik tombol <i>send</i> 	<ul style="list-style-type: none"> token 	<ul style="list-style-type: none"> status <i>message</i> data laporan harga token

b Email

ID Pengujian	Deskripsi Pengujian	Prosedur Pengujian	Data Masukan	Keluaran yang diharapkan (JSON)
Em.1	Mengirim email lupa <i>password</i>	<ul style="list-style-type: none"> Memilih <i>method</i> POST Klik tombol <i>send</i> 	<ul style="list-style-type: none"> Username 	<ul style="list-style-type: none"> status <i>message</i> data pengguna
Em.2	Mengirim email verifikasi <i>account</i>	<ul style="list-style-type: none"> Memilih <i>method</i> POST Klik tombol <i>send</i> 	<ul style="list-style-type: none"> Username 	<ul style="list-style-type: none"> status <i>message</i> data pengguna
Em.3	Mengirim ulang email verifikasi <i>account</i>	<ul style="list-style-type: none"> Memilih <i>method</i> POST Klik tombol <i>send</i> 	<ul style="list-style-type: none"> Username 	<ul style="list-style-type: none"> status <i>message</i> data pengguna

Lampiran 12 *Acceptance test criteria* iterasi ketiga (lanjutan)

ID Pengujian	Deskripsi Pengujian	Prosedur Pengujian	Data Masukan	Keluaran yang diharapkan (JSON)
Em.4	Memverifikasi <i>account</i>	<ul style="list-style-type: none"> Memilih <i>method</i> GET Klik tombol <i>send</i> 	<ul style="list-style-type: none"> token 	<ul style="list-style-type: none"> status <i>message</i> data pengguna
Em.5	Memperbarui <i>password</i>	<ul style="list-style-type: none"> Memilih <i>method</i> POST Klik tombol <i>send</i> 	<ul style="list-style-type: none"> token <i>password</i> baru 	<ul style="list-style-type: none"> status <i>message</i> data pengguna

c Operasi pasar

ID Pengujian	Deskripsi Pengujian	Prosedur Pengujian	Data Masukan	Keluaran yang diharapkan (JSON)
Op.10	Memberikan tanggapan operasi pasar	<ul style="list-style-type: none"> Memilih <i>method</i> POST Klik tombol <i>send</i> 	<ul style="list-style-type: none"> token operasiPasar_id isi 	<ul style="list-style-type: none"> status <i>message</i> data operasi pasar token
Op.11	Menghapus tanggapan operasi pasar	<ul style="list-style-type: none"> Memilih <i>method</i> POST Klik tombol <i>send</i> 	<ul style="list-style-type: none"> token operasiPasar_id _id 	<ul style="list-style-type: none"> status <i>message</i> data operasi pasar token
Op.12	Melihat tanggapan operasi pasar	<ul style="list-style-type: none"> Memilih <i>method</i> GET Klik tombol <i>send</i> 	<ul style="list-style-type: none"> token operasiPasar_id 	<ul style="list-style-type: none"> status <i>message</i> data operasi pasar token

Lampiran 12 *Acceptance test criteria* iterasi ketiga (lanjutan)

Lokasi

ID Pengujian	Deskripsi Pengujian	Prosedur Pengujian	Data Masukan	Keluaran yang diharapkan (JSON)
Lk.1	Melihat seluruh provinsi di Indonesia	<ul style="list-style-type: none"> Memilih <i>method</i> GET Klik tombol <i>send</i> 	<ul style="list-style-type: none"> provinsi 	<ul style="list-style-type: none"> status <i>message</i> data provinsi
Lk.2	Melihat kabupaten yang berada pada suatu provinsi	<ul style="list-style-type: none"> Memilih <i>method</i> GET Klik tombol <i>send</i> 	<ul style="list-style-type: none"> kabupaten id_prov 	<ul style="list-style-type: none"> status <i>message</i> data kabupaten
Lk.3	Melihat kecamatan yang berada pada suatu kabupaten	<ul style="list-style-type: none"> Memilih <i>method</i> GET Klik tombol <i>send</i> 	<ul style="list-style-type: none"> kecamatan id_kab 	<ul style="list-style-type: none"> status <i>message</i> data kecamatan
Lk.4	Melihat kelurahan yang ada pada suatu kecamatan	<ul style="list-style-type: none"> Memilih <i>method</i> GET Klik tombol <i>send</i> 	<ul style="list-style-type: none"> kelurahan id_kec 	<ul style="list-style-type: none"> status <i>message</i> data kelurahan

RIWAYAT HIDUP

Penulis dilahirkan di Sragen pada tanggal 17 Agustus 1996 dari pasangan dari Sunardi dan Rusmiyanti. Penulis adalah putra pertama dari dua bersaudara. Pada tahun 2013, penulis lulus dari SMA Negeri 1 Sragen dan diterima di Departemen Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Pertanian Bogor melalui jalur SNMPTN Undangan di tahun yang sama.

Pada tahun 2016, penulis melaksanakan Praktik Kerja Lapangan (PKL) di Komunitas Mandiri Indonesia (KMI) Bogor selama 35 hari kerja. Pada masa perkuliahan, penulis aktif menjadi asisten praktikum mata kuliah Rekayasa Perangkat Lunak.