

**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 DISERTASI 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 disertasi ini.

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

Bogor, Juli 2017

Nugroho Agung Cahyono
NIM G64130039

ABSTRAK

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

Kesenjangan informasi yang terjadi antara petani dengan pemerintah merupakan permasalahan yang terjadi dalam mengelola dan mengembangkan sektor pertanian. Kesenjangan informasi tersebut menyebabkan fluktuasi harga. Perkembangan internet dapat dijadikan sebagai peluang untuk mengatasi kasus fluktuasi tersebut 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 sehingga stabilitas harga dapat tercapai. Penelitian ini mengembangkan modul pelaporan harga menggunakan REST API dengan metode *extreme programming*. Metode *extreme programming* merupakan bagian dari *agile software engineering*. Basis data yang digunakan, yaitu basis data non relasional.

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

ABSTRACT

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

The informations gap between farmers and the government is a problem that occurs in managing and developing the agricultural sector. The information gap causes price fluctuations. The development of the internet can serve as a chance to overcome the fluctuation case by providing media that can connect extension workers, farmers, communities, traders, and government. Goal of this study to provide information and communication media between government, farmers, communities, and traders so that price stability can be achieved. This research develops price reporting module using REST API with extreme programming method. The extreme programming method is part of agile software engineering. The database used is non-relational database.

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 Ilmu Komputer
pada
Departemen Ilmu Komputer

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

Penguji:

- 1 Rina Trisminingsih, S.Komp, MT
- 2 Husnul Khotimah, S.Komp, M.Kom

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:

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, 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 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, Juli 2017

Nugroho Agung Cahyono

DAFTAR ISI

DAFTAR TABEL	vi
DAFTAR GAMBAR	vi
DAFTAR LAMPIRAN	vi
PENDAHULUAN	1
Latar Belakang	1
Perumusan Masalah	2
Tujuan Penelitian	2
Manfaat Penelitian	2
Ruang Lingkup Penelitian	2
TINJAUAN PUSTAKA	3
JSON (Java Script Object Notation)	3
REST API	3
MongoDB	3
Roadmap Penelitian	3
METODE	4
Data Penelitian	4
Arsitektur Penelitian	5
Tahapan Penelitian	5
Lingkungan Pengembangan	7
HASIL DAN PEMBAHASAN	7
Tahapan Penelitian	7
SIMPULAN DAN SARAN	24
Simpulan	24
Saran	24
DAFTAR PUSTAKA	25
LAMPIRAN	27
RIWAYAT HIDUP	55

DAFTAR TABEL

1	<i>User story</i> masyarakat	8
2	Hasil pengujian iterasi pertama	13
3	Hasil pengujian iterasi kedua	18
4	Hasil pengujian iterasi ketiga	23

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	9
5	<i>CRC cards</i> iterasi pertama	10
6	<i>Class diagram</i> iterasi pertama	10
7	Daftar <i>collections</i> di <i>basis data</i> iterasi pertama	11
8	<i>Collections</i> komoditas iterasi pertama	11
9	<i>Sequence diagram</i> fungsi tambah laporan harga	12
10	Pengujian fungsi melihat laporan harga	14
11	<i>CRC cards</i> iterasi kedua	15
12	<i>Class diagram</i> iterasi kedua	16
13	<i>Sequence diagram</i> fungsi <i>forgetPassword</i>	16
14	Potongan <i>code</i> fungsi <i>getDayLaporan</i>	17
15	Pengujian fungsi melihat laporan harga 20 hari sebelumnya	18
16	<i>CRC cards</i> iterasi ketiga	20
17	<i>Class diagram</i> iterasi ketiga	21
18	<i>Code</i> memasukkan data format JSON ke <i>basis data</i> <i>mongoDB</i>	21
19	Integrasi fungsi <i>getMailVerify</i> dengan fungsi <i>register</i>	21
20	Pengujian fungsi melihat provinsi di Indonesia	22
21	Dokumentasi API di Swaggerhub	23

DAFTAR LAMPIRAN

1	Deskripsi <i>user story</i>	27
2	Daftar <i>collections</i> iterasi pertama	28
3	<i>Sequence diagram</i> iterasi pertama	29
4	<i>Acceptance test criteria</i> iterasi pertama	37
5	<i>Sequence diagram</i> iterasi kedua	40
6	<i>Acceptance test criteria</i> iterasi kedua	43
7	<i>Sequence diagram</i> iterasi ketiga	45
8	<i>Acceptance test criteria</i> iterasi ketiga	52

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.

Kesenjangan informasi yang terjadi antara petani dengan pemerintah merupakan permasalahan yang terjadi dalam mengelola dan mengembangkan sektor pertanian. Kesenjangan tersebut dimanfaatkan oleh pihak ketiga, yaitu tengkulak yang membeli hasil panen dari petani dengan harga rendah dan dijual kembali dengan harga tinggi. Kondisi tersebut menimbulkan 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 tidak stabil membuat petani merasa kesulitan untuk memasarkan hasil pertaniannya. Kondisi tersebut disebabkan oleh jarak sumber informasi yang jauh, sehingga informasi yang diterima oleh petani tidak *up-to-date* (Saputra 2015).

Harga komoditas pertanian yang tidak stabil membuat pedagang kesulitan untuk menentukan harga jual yang sesuai. Harga yang tidak stabil 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 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 2014).

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. Penelitian yang dilakukan oleh Ahmad (2015) tentang perancangan aplikasi komoditas pertanian berbasis Android. Penelitian ini melanjutkan penelitian sebelumnya yang dilakukan oleh Setyatama (2016), yaitu pengembangan aplikasi pelaporan harga komoditas pertanian berbasis *mobile* menggunakan REST API. Penelitian tersebut perlu pengembangan token OAuth

menjadi token yang dinamis untuk keamanan data (Setyatama 2016). Selain itu penelitian sebelumnya yang dilakukan oleh Mardika (2016) tentang pengembangan modul pelaporan harga komoditas pertanian pada aplikasi portal harga untuk petani menggunakan REST API belum terhubung dengan media sosial. Penelitian ini mengembangkan modul pelaporan harga dengan metode *extreme programming*. Modul yang dikembangkan mampu menjadi media informasi antara *stakeholder* terkait dan mampu menyediakan informasi seputar harga komoditas pertanian kepada *stakeholder* terkait secara *up-to-date*.

Perumusan Masalah

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

Tujuan Penelitian

Penelitian ini bertujuan mengembangkan modul pelaporan harga komoditas pertanian menggunakan REST API dan basis data non relasional, sehingga menghubungkan antar *stakeholder* terkait agar saling terintegrasi.

Manfaat Penelitian

Manfaat dari penelitian ini adalah:

1. Dari sisi pedagang
 - a Informasi harga komoditas pertanian bisa diketahui secara *real-time*.
 - b Mampu menentukan harga jual komoditas pertanian.
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 Mampu mengendalikan harga komoditas di pasar.
 - b Mengetahui daerah yang membutuhkan operasi pasar.

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 pelaporan harga.

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 pemrograman apapun. JSON dirancang untuk memudahkan pertukaran data dan merupakan perluasan dari fungsi-fungsi *Javascript* (Herdiana 2014).

REST API

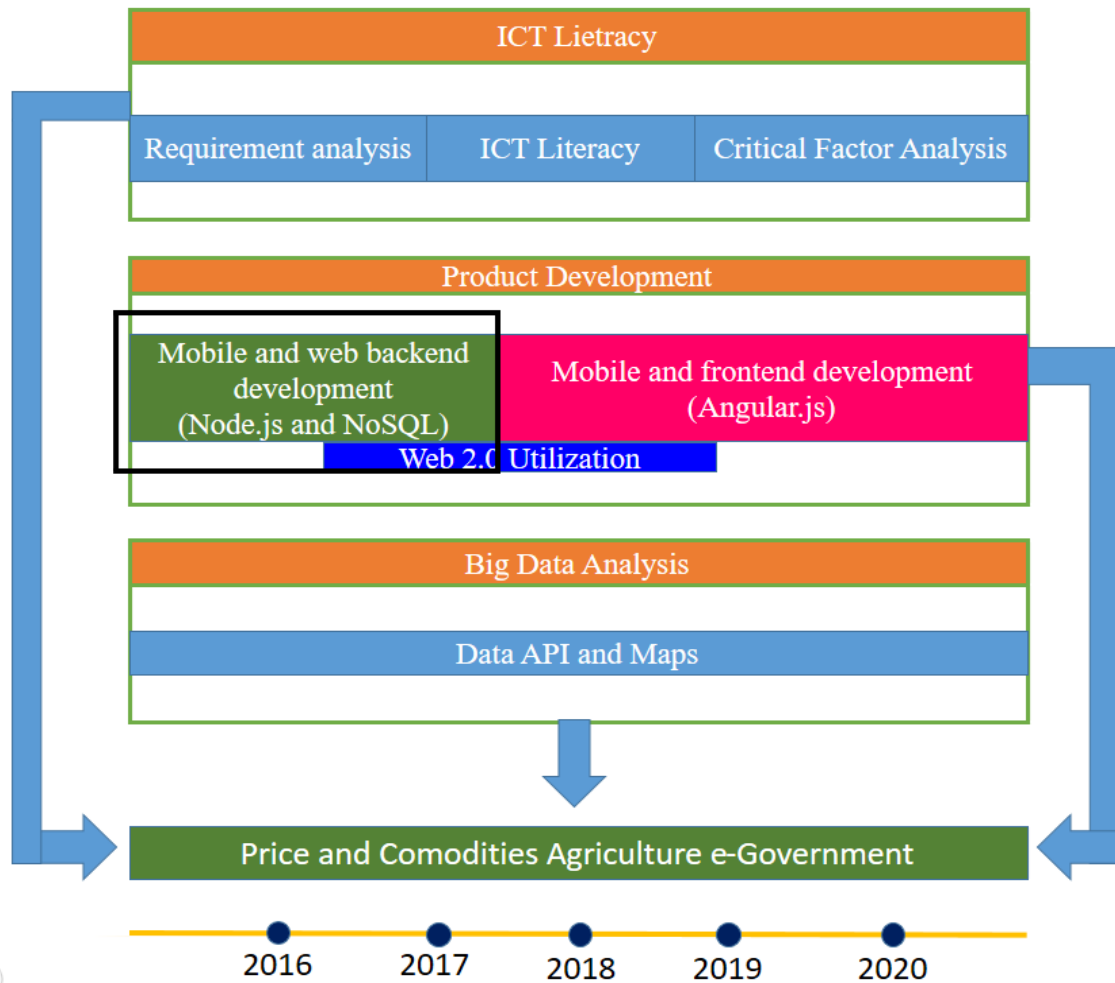
REST (*Representational State Transfer*) merupakan sebuah arsitek untuk aplikasi *web services*, dirancang untuk menggunakan protokol HTTP (*Hypertext Transfer Protocol*) sebagai penghubung komunikasi antara mesin dan mesin, tanpa menggunakan mekanisme yang kompleks. Perintah HTTP yang bisa digunakan 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 merupakan salah satu jenis basis data non relasional dengan tipe *document store basis data*. MongoDB dapat digunakan untuk menyimpan 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 *et al.* 2016).

Roadmap Penelitian

Roadmap merupakan perencanaan dari seluruh pekerjaan yang akan dikerjakan secara detail 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* (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, yaitu 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. Basis data yang digunakan adalah basis data non relasional.

METODE

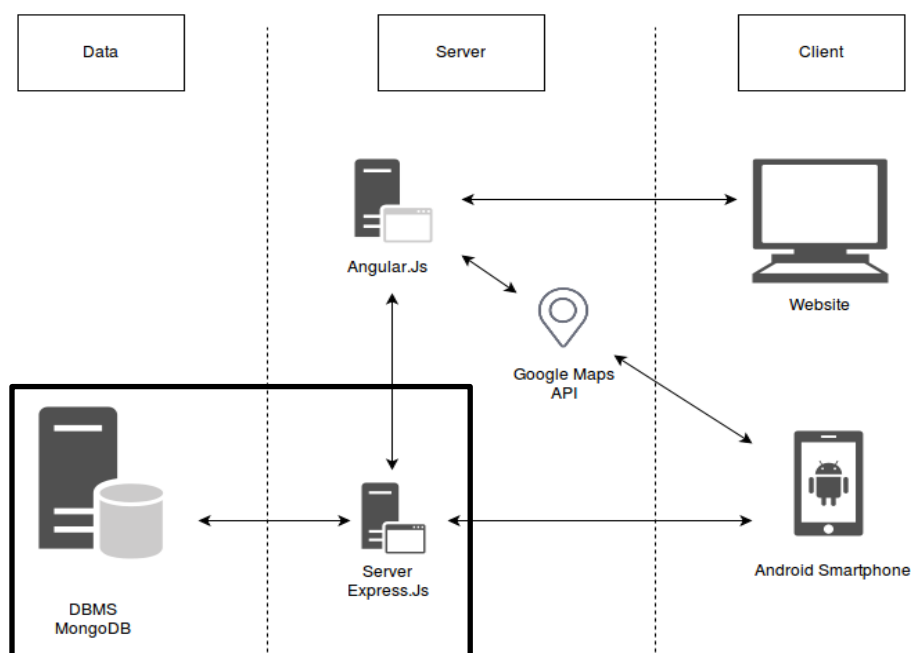
Data Penelitian

Data yang digunakan antara lain, yaitu komoditas, laporan harga, operasi pasar, dan lokasi. Data komoditas terdiri atas nama komoditas, satuan, dan harga yang ditentukan oleh pemerintah. Data laporan harga terdiri atas jenis komoditas, harga, alamat, *latitude* dan *longitude* saat masyarakat melaporkan harga. Data

operasi pasar terdiri atas jenis komoditas, lokasi, dan pesan yang berasal dari masyarakat. Data lokasi terdiri dari data provinsi, kabupaten, kecamatan, dan kelurahan.

Arsitektur Penelitian

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

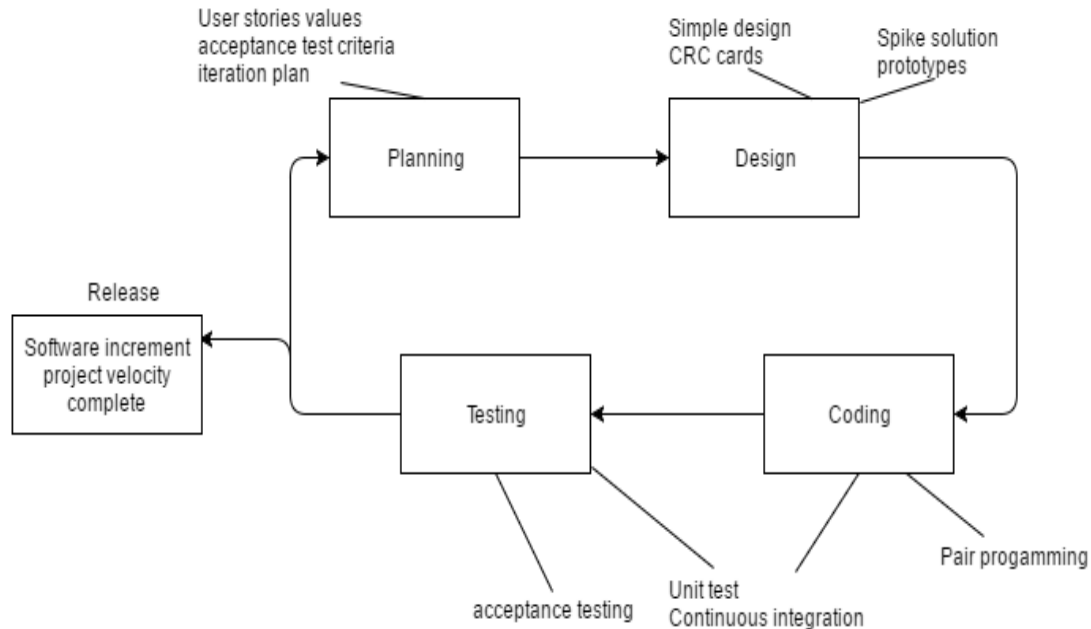


Gambar 2 Arsitektur Penelitian

Tahapan Penelitian

Metode yang digunakan adalah *extreme programming* yang merupakan bagian dari metode *agile software development* yang berbasis pada pengembangan iteratif dan kolaborasi antar anggota tim yang terorganisir (Rohman 2015). *Extreme programming* merupakan metodologi pengembangan perangkat lunak yang tanggap terhadap perubahan kebutuhan pengguna, sehingga meningkatkan kualitas perangkat lunak (Pressman 2010). Perubahan *requirements* dari pengguna dapat

segera ditanggapi oleh pengembang meskipun pengembangan perangkat lunak sudah dilakukan (Oktaviani dan Hutrianto 2016). Tahapan dalam *extreme programming*, yaitu *planning*, *design*, *coding*, dan *testing* dapat dilihat pada Gambar 3.



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

1 Perencanaan (*Planning*)

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. Iterasi yang akan dilakukan selama penelitian sebanyak tiga kali.

2 Desain (*Design*)

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

3 Pengkodean (*Coding*)

Tahap pengkodean 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 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 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
- *Test API*: Postman 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.

Iterasi Pertama

Perencanaan (*Planning*)

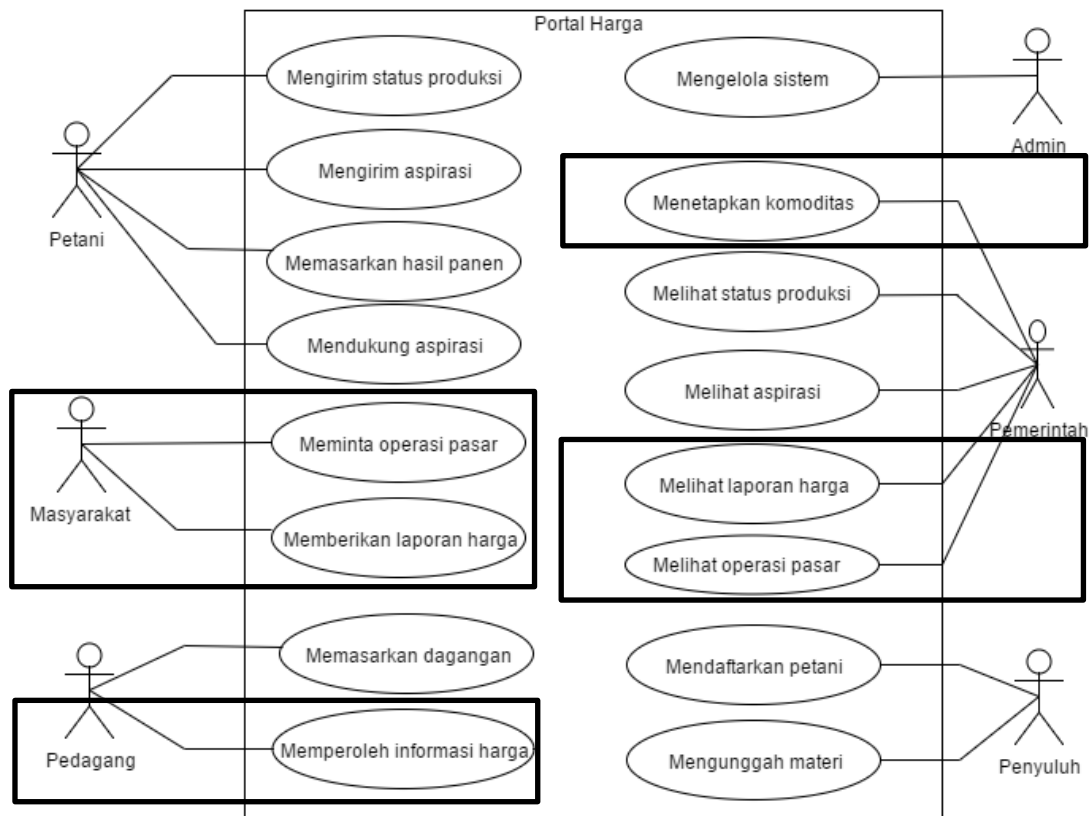
Perencanaan awal dilakukan oleh tim untuk menentukan *stakeholder* dan *role* dari setiap *stakeholder*. Berdasarkan hasil diskusi dengan tim, terdapat enam *stakeholder*, yaitu *admin*, pemerintah, penyuluh, petani, masyarakat, dan pedagang. *Role* setiap *stakeholder* sebagai berikut: *role* 1 adalah *admin*, *role* 2 adalah pemerintah, *role* 3 adalah penyuluh, *role* 4 adalah petani, *role* 5 adalah masyarakat, dan *role* 6 adalah pedagang.

Membuat modul dan fungsi-fungsi dasar berdasarkan kebutuhan fungsional *stakeholder* yang sudah ditentukan. Modul yang dibuat 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*, yaitu *user story* masyarakat. Deskripsi *user story* masyarakat dapat dilihat pada tabel 1. Detail deskripsi *user story* dapat dilihat *user story* deskripsi 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. *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 1 *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.

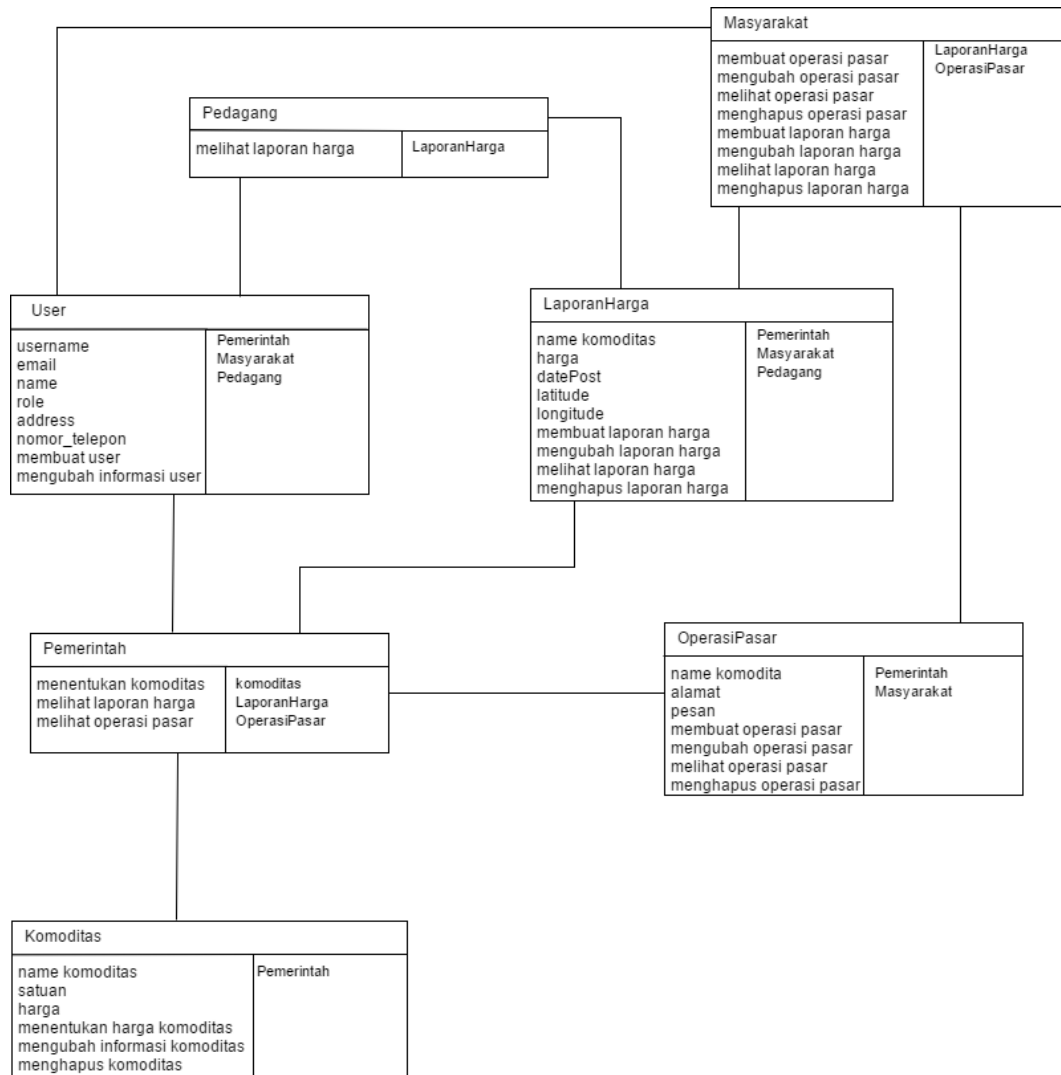


Gambar 4 Use case diagram iterasi pertama

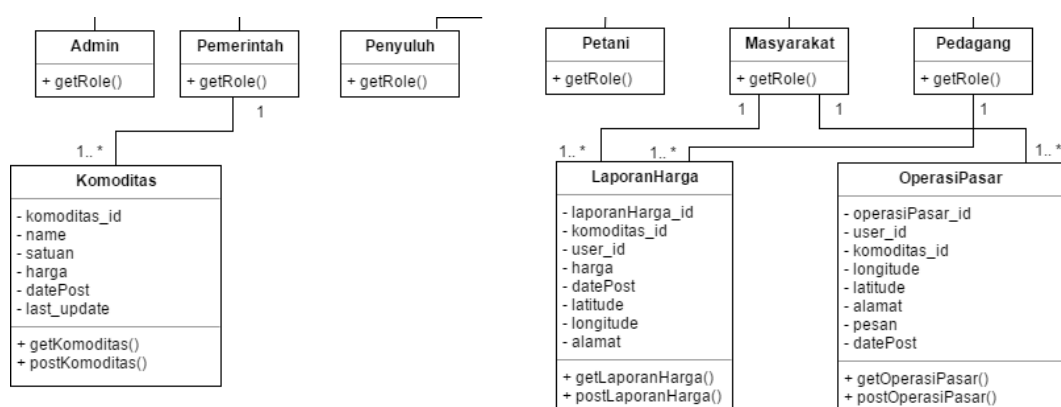
Desain (Design)

Merancang CRC cards yang akan 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*. *Class* komoditas hanya dapat diakses oleh pemerintah, namun pemerintah dapat mengakses *class* laporan harga maupun *class* operasi pasar. *Class* laporan harga dan operasi pasar dapat diakses oleh *class* masyarakat. *Class* pedagang hanya dapat mengakses *class* laporan harga. *Class diagram* diimplementasikan pada basis data non relasional mongoDB dengan setiap *class* yang dibuat akan disimpan dalam sebuah *collections* di dalam *basis data*. Gambar daftar *collections* dapat dilihat pada Gambar 6.

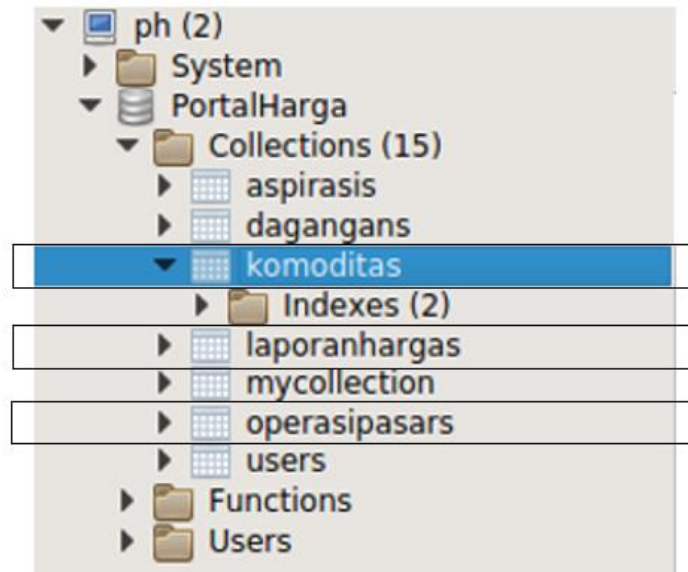
Class diagram iterasi pertama terdiri dari *class* komoditas, laporan harga, dan operasi pasar yang diimplementasikan menjadi *collections* di *database* non relasional dapat dilihat pada Gambar 7. *Collections* komoditas memiliki *field* komoditas_id, name, satuan, harga, datePost, dan last_update yang dapat dilihat pada Gambar 8. Komoditas_id merupakan *filed primary key* yang bersifat *auto increment*. Name, satuan, dan harga merupakan nama komoditas, satuan komoditas, dan harga komoditas yang berasal dari masukkan yang dilakukan oleh pemerintah. Sedangkan datePost dan last_update merupakan waktu saat memasukkan data dan memperbarui data komoditas. Daftar *collections* iterasi pertama dapat dilihat pada lampiran 2.



Gambar 5 CRC cards iterasi pertama



Gambar 6 Class diagram iterasi pertama



Gambar 7 Daftar *collections* di *basis data* iterasi pertama

```

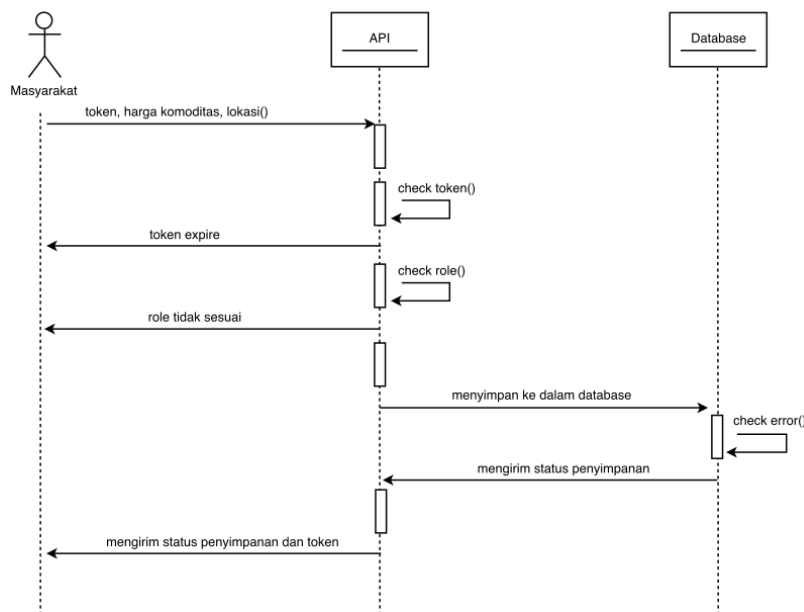
9   var komoditasModel = new Schema({
10     name:String,
11     satuan:String,
12     harga:Number,
13     datePost:Number,
14     last_update:Number
15   });
16
17   komoditasModel.plugin(autoIncrement.plugin,{model:'komoditas',
18     field:'komoditas_id',startAt:1});

```

Gambar 8 *Collections* komoditas iterasi pertama

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*. *Class admin*, pemerintah, penyuluh, petani, masyarakat, dan pedagang merupakan turunan dari *class user*.

Tahap desain juga dilakukan perencanaan dan pembuatan *sequence diagram*. Salah satu *sequence diagram*, yaitu fungsi tambah laporan pada modul laporan harga yang dapat dilihat pada Gambar 9. Alur pekerjaan yang akan dilakukan adalah menambah laporan harga, masyarakat memasukkan data harga suatu komoditas, lokasi, dan token ke API, token yang dikirim akan dicek oleh API. Berdasarkan token API juga mengecek *role* pengguna, jika token dan *role* sesuai maka API akan menyimpan data ke dalam basis data dan basis data akan mengecek apakah ada *error*, API menerima status penyimpanan apakah berhasil atau *error* dari basis data dan mengirimkannya ke masyarakat. Detail *sequence diagram* iterasi pertama dapat dilihat pada Lampiran 3.



Gambar 9 *Sequence diagram* fungsi tambah laporan harga

Pengkodean (Coding)

Method yang digunakan untuk mengakses REST API, yaitu POST dan GET. *Port* yang digunakan untuk mengakses REST API adalah 5000. Penggunaan *method* POST dan GET diatur di dalam *route*. *Method* POST digunakan untuk memasukkan data ke dalam *basis data*, sedangkan *method* GET untuk memperoleh data dari *basis data*. *Method* POST data yang akan dimasukkan diletakkan di *body*. Format keluaran setelah mengakses API dengan *method* POST atau GET terdiri dari JSON *array* yang diawali dan diakhiri dengan kurung siku (“[]”) dan JSON *object* yang diawali dan diakhiri dengan kurung kurawal (“{}”). 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 sukses, 204 berarti data kosong, 401 berarti *unauthorized* yang disebabkan oleh token yang tidak valid, dan 408 berarti *request timeout*.

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

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

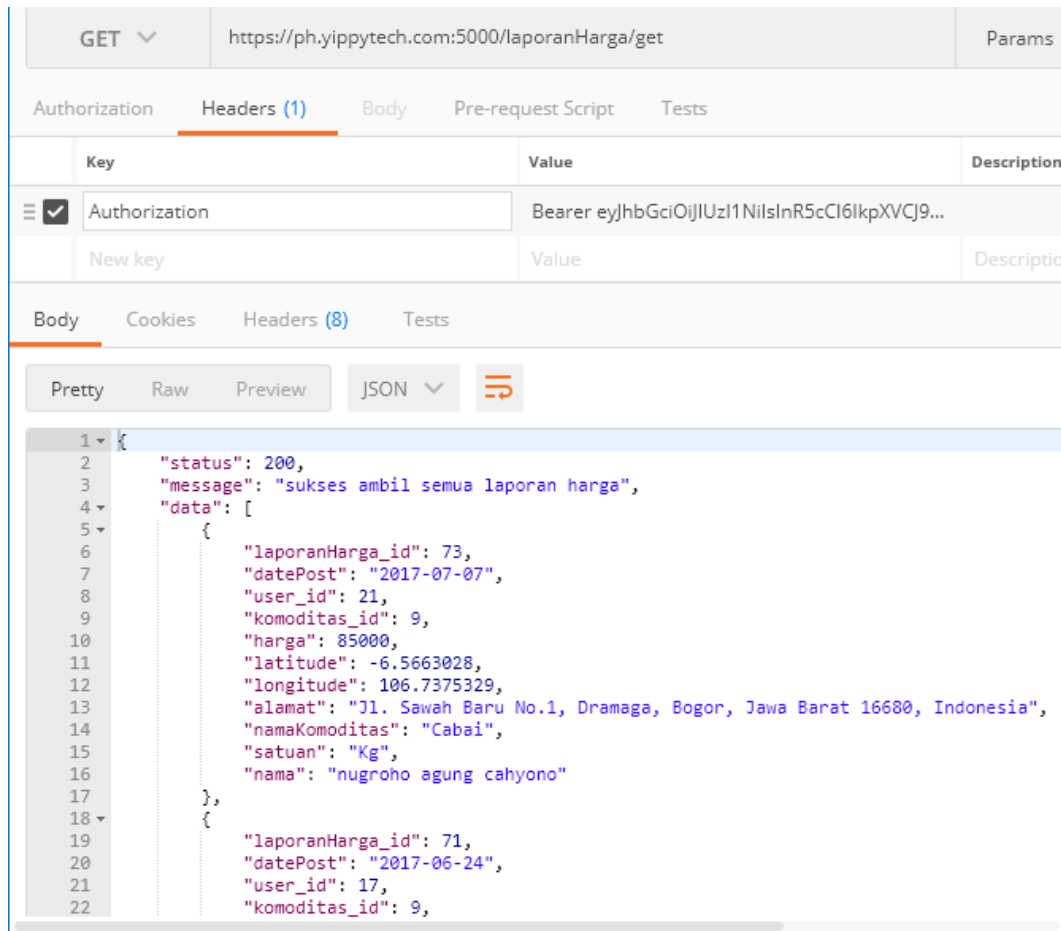
Pengujian (Testing)

Pengujian yang dilakukan, yaitu pengujian fungsi *create*, *read*, *update*, dan *delete* untuk setiap modul. Pengujian pertama adalah pengujian token, mengecek ketersediaan token dan memverifikasi token yang diberikan. Pengujian selanjutnya, yaitu mengecek dan memverifikasi *role* yang berasal dari hasil *decode* token. *Role* digunakan untuk mengakses suatu fungsi tertentu. Pengujian terakhir berhubungan dengan basis data, yaitu mengambil data dari basis data, jika data yang diminta tidak tersedia pada basis data atau kosong, maka gagal menyimpan data ke dalam basis data, dan parameter yang diberikan tidak sesuai yang ada di basis data. Pengujian pada iterasi pertama dilakukan sesuai dengan *acceptance test criteria* iterasi pertama yang dapat dilihat pada Lampiran 4. Hasil pengujian pada iterasi pertama dapat dilihat pada Tabel 2.

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.

Tabel 2 Hasil pengujian iterasi pertama

<i>Class</i>	<i>Method</i>	Fungsi	URL	Status
Komoditas	POST	Menambah komoditas	komoditas/add	Berhasil
	GET	Melihat komoditas	komoditas/get	Berhasil
	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
	POST	Menghapus operasi pasar	operasiPasar/delete	Berhasil



Gambar 10 Pengujian fungsi melihat laporan harga

Iterasi Kedua

Perencanaan (*Planning*)

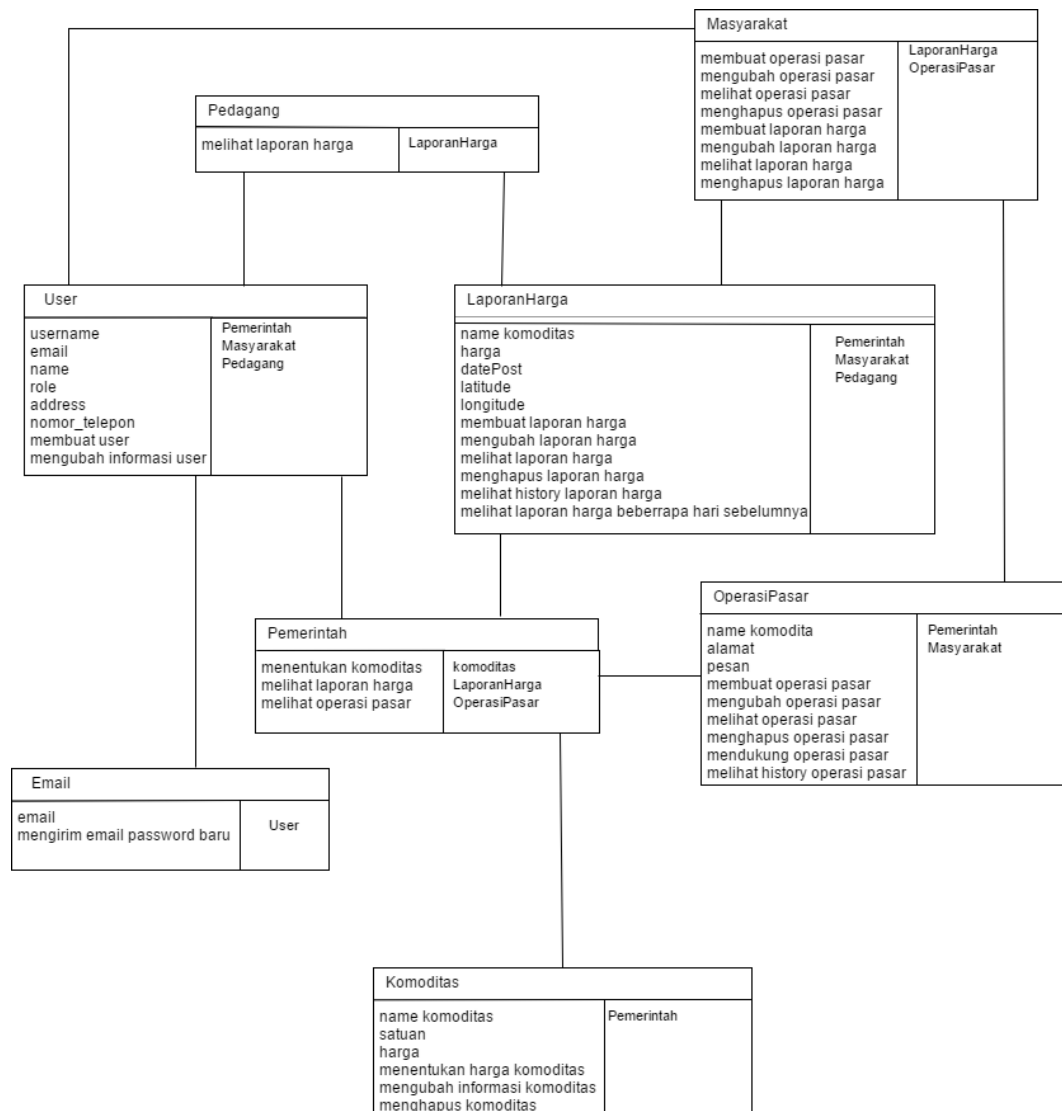
Modul operasi pasar, laporan harga dikembangkan. Pada modul operasi pasar pengguna dapat mendukung operasi pasar lain. Pada modul laporan harga dapat melihat laporan harga pada beberapa hari sebelumnya. Modul operasi pasar dan laporan harga dapat melihat *history* pengguna. Selain itu penelitian sebelumnya yang dilakukan oleh Mahardika (2016) belum terintegrasi dengan media sosial, maka modul email dikembangkan untuk mengatasi kebutuhan yang lebih umum. Modul email dikembangkan untuk mengatasi pengguna yang lupa *password*.

Desain (*Design*)

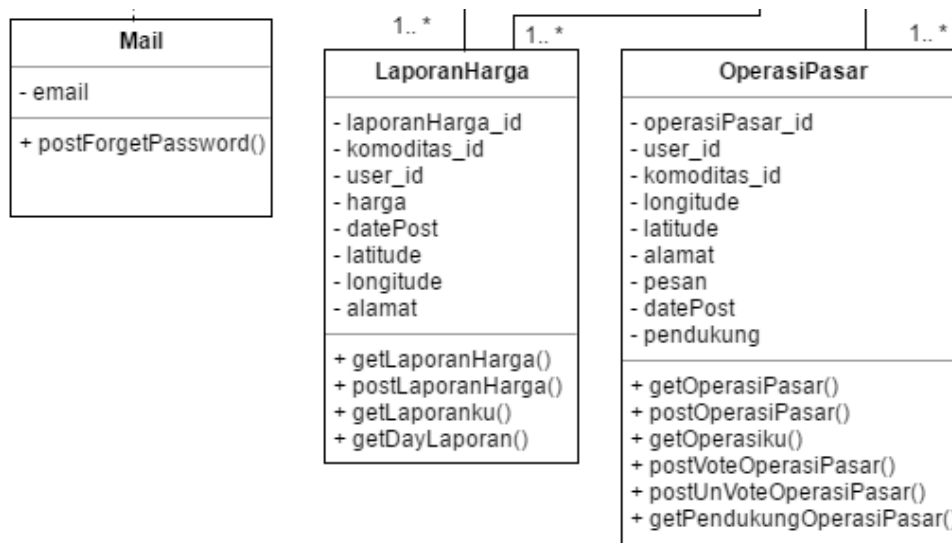
Merancang CRC cards yang akan menjadi *class diagram* sesuai dengan modul pada tahap perencanaan yang dapat dilihat pada Gambar 11. Perancangan *class diagram* sesuai dengan CRC cards yang dibuat. Pembuatan *sequence diagram* disesuaikan dengan fungsi yang ditambahkan pada *class diagram*.

Class operasi pasar terdapat empat fungsi yang dikembangkan, yaitu fungsi *vote*, *unvote*, *getPendukung*, dan *operasiku* yang dapat dilihat pada Gambar 12. 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

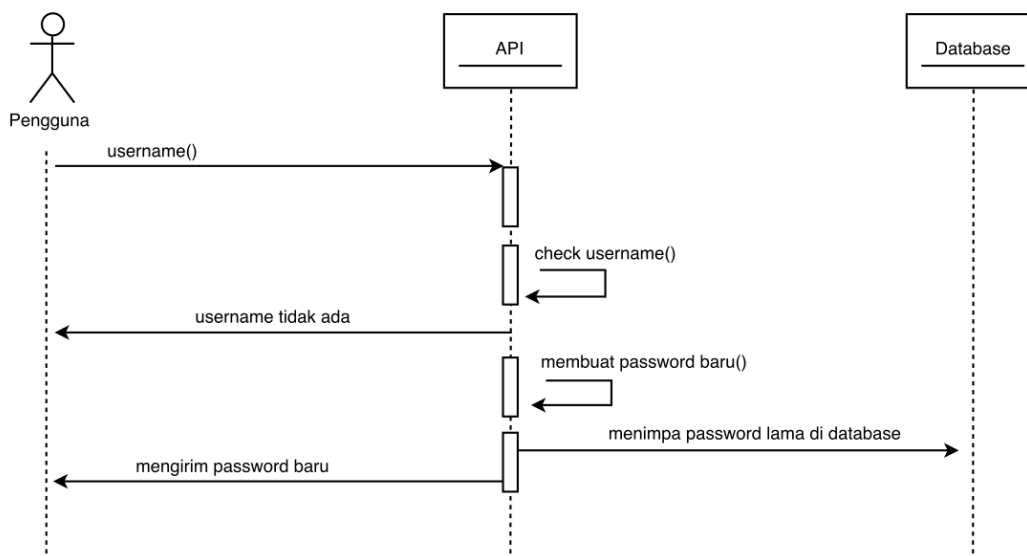
harga terdapat dua fungsi yang dikembangkan, yaitu fungsi laporanku dan getDayLaporan. Fungsi laporanku untuk melihat *history* laporan harga pengguna. Fungsi getDayLaporan untuk mendapatkan laporan harga beberapa hari sebelumnya. *Class* email dikembangkan satu fungsi yaitu forgetPassword. Fungsi tersebut akan memberikan *password* baru kepada pengguna yang dikirimkan melalui email.



Gambar 11 CRC cards iterasi kedua

Gambar 12 *Class diagram* iterasi kedua

Salah satu *sequence diagram* adalah fungsi `forgetPassword` yang dapat dilihat pada Gambar 13. Pengguna memasukkan *username* kemudian API akan membuat *password* baru untuk pengguna. API akan menimpa *password* lama pengguna yang ada di basis data. API akan mengirimkan *password* baru kepada pengguna melalui email. Detail *sequence diagram* iterasi kedua dapat dilihat pada Lampiran 5.

Gambar 13 *Sequence diagram* fungsi `forgetPassword`

Pengkodean (Coding)

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.

Tahap pengkodean 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*.

```

221 laporanHarga.find({'-id -_v'}, {sort: {datePost: -1}}, function(err, all){
222   if(all==null){
223     res.json({status:204,message:err,data:"",token:req.token});
224   }else{
225     //tanggal sekarang
226     var dateNow = new Date();
227     //tanggal sekarang di kurangi hari yang diinginkan, hari nya
228     dateNow.setDate(dateNow.getDate() - req.params.day);
229     //hari yang diinginkan dalam format, hari, tanggal, bulan, dan tahun
230     var getDate = dateFormat(dateNow, "dddd , mmmm dS , yyyy");
231     //console.log(getDate);
232     //buat variabel parsing yang akan menerima laporanHarga_id pada hari itu
233     var parsing = [];
234     var number = [];
235     var counter = 0;
236
237     for(var i=0;i<all.length;i++){
238       if(dateFormat(all[i].datePost, "dddd , mmmm dS , yyyy")==getDate){
239         number.push(all[i].laporanHarga_id);
240       }
241     }
242     //time out 300 miliseconds
243     setTimeout(function () {
244       for(var i=0;i<number.length;i++){
245         laporanHarga.findOne({laporanHarga_id:number[i]}, {'-id -_v'}).lean().exec(function(err, laporan){
246           komoditas.findOne({komoditas_id:laporan.komoditas_id}).exec(function(err, komo){
247             laporan.namaKomoditas=komo.name;
248             laporan.satuan = komo.satuan;
249             laporan.datePost = moment(laporan.datePost).format("YYYY-MM-DD");
250             parsing.push(laporan);
251           })
252         })
253       }
254     }, 300);
255     //time out 400 mili seconds
256     setTimeout(function () {
257       res.json({
258         status:200,
259         message:"sukses mendapat laporan harga " + req.params.day + ' sebelumnya',
260         data:parsing,
261         token:req.token
262       });
263     }, 400);

```

Gambar 14 Potongan *code* fungsi `getDayLaporan`

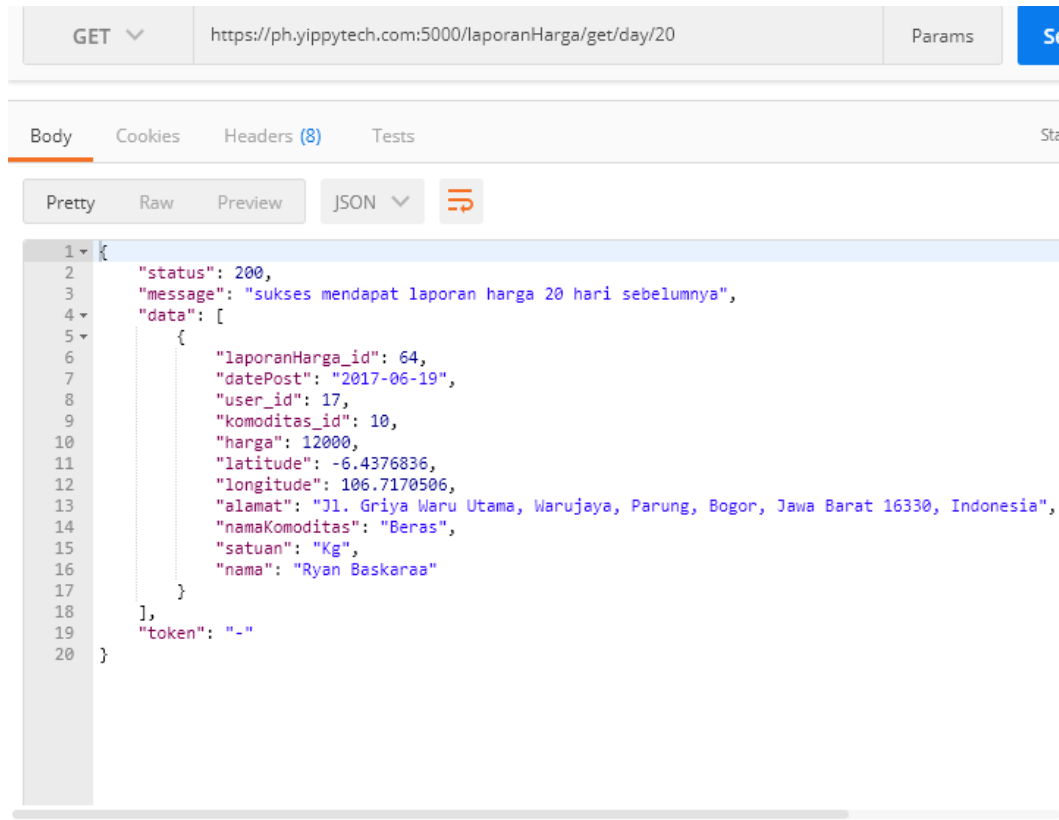
Implementasi modul email menggunakan *nodemailer* yang merupakan Gmail Google. Fungsi `forgetPassword` ketika diakses akan membuat *password* baru dan memperbarui *password* pengguna yang lama. *Password* baru dikirimkan ke alamat email pengguna yang merupakan hasil *random string*. *Random string* yang 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 6. Contoh pengujian, yaitu fungsi `forgetPassword` yang dilakukan dengan *login* menggunakan *password* baru yang dikirimkan melalui email. Hasil pengujian pada iterasi kedua dapat dilihat pada Tabel 3.

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 20 hari sebelumnya dengan mengakses URL <https://ph.yippytech.com:5000/laporanHarga/get/day/20> menggunakan *method* GET. Angka 20 merupakan sebuah parameter berapa hari sebelumnya yang dapat diganti sesuai dengan kebutuhan. Format keluaran berupa JSON yang terdiri atas *status*, *message*, *data*, dan *token*. *Data* berisi laporan harga

yang dilaporkan 20 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.



Gambar 15 Pengujian fungsi melihat laporan harga 20 hari sebelumnya

Tabel 3 Hasil pengujian iterasi kedua

Class	Method	Fungsi	URL	Status
Laporan harga	POST	Mendukung operasi pasar	operasiPasar/pendukung/add	Berhasil
	POST	Membatalkan dukungan operasi pasar	operasiPasar/pendukung/delete	Berhasil
	GET	Melihat pendukung operasi pasar	operasiPasar/pendukung/get/:operasiPasar_id	Berhasil
	GET	Melihat <i>history</i> laporan harga	operasiPasar/operasi/get/:user_id	Berhasil
Operasi pasar	GET	Melihat <i>history</i> operasi pasar	laporanHarga/get/laporan/:user_id	Berhasil

<i>Class</i>	<i>Method</i>	Fungsi	URL	Status
	GET	Melihat laporan harga beberapa hari sebelumnya	laporanHarga/get/day/:day	Berhasil
Mail	POST	Mengirim password	email/forgetPassword	Berhasil

Iterasi Ketiga

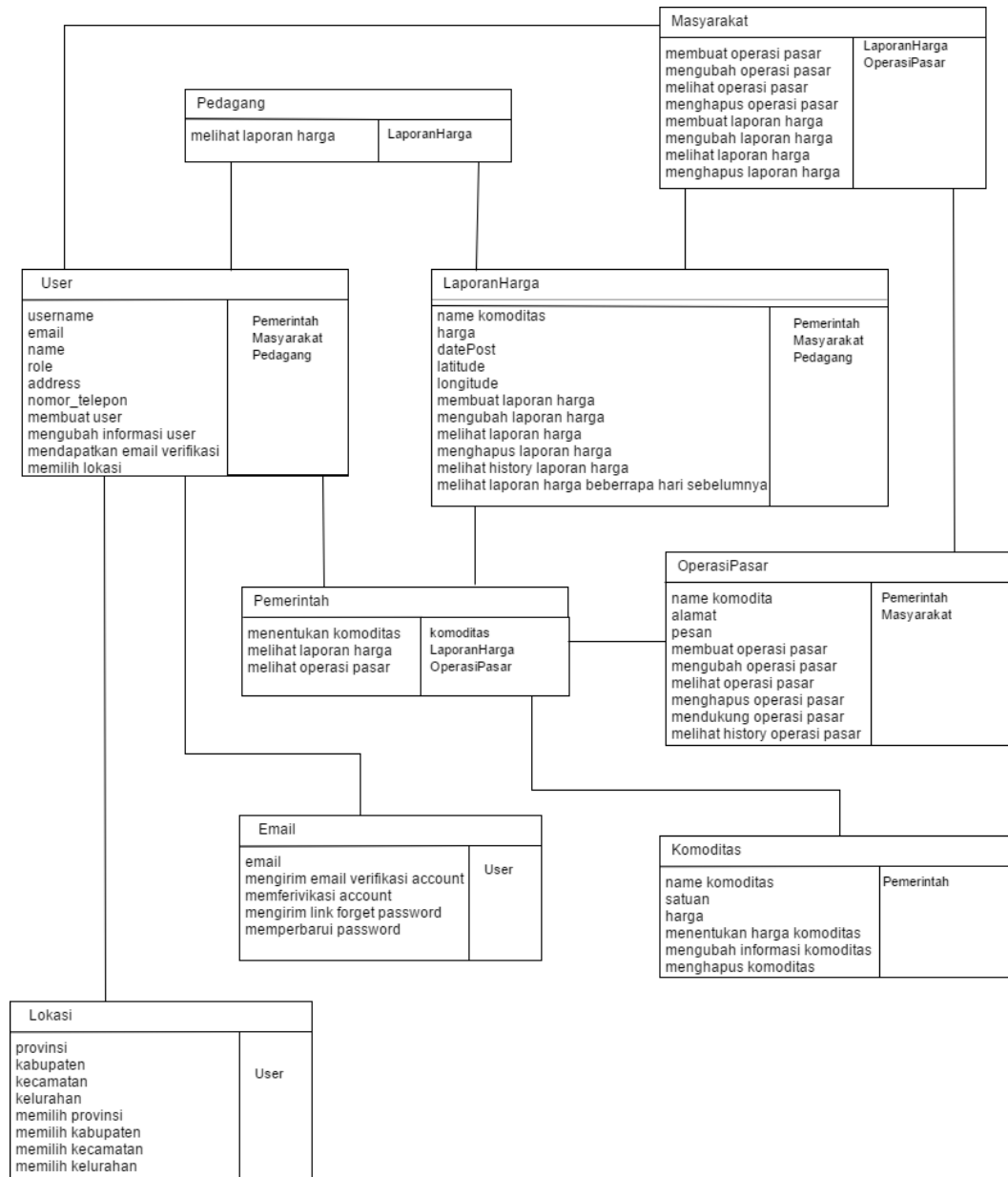
Perencanaan (*Planning*)

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. 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*.

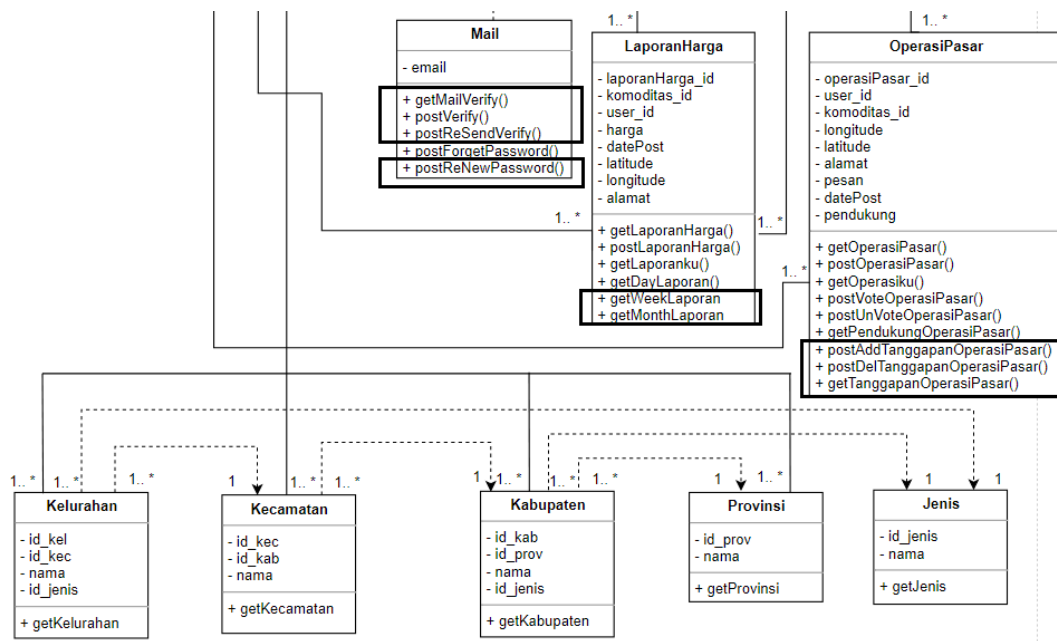
Desain (*Design*)

Merancang CRC *cards* yang akan menjadi *class diagram* sesuai modul yang dikembangkan pada tahap perencanaan yang dapat dilihat pada gambar 16. Selain itu, pada tahap desain juga dilakukan perencanaan dan pembuatan *sequence diagram*. Detail *sequence diagram* iterasi ketiga dapat dilihat pada Lampiran 7.

Modul lokasi terdiri dari *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. *Class diagram* iterasi ketiga secara detail dapat dilihat pada Gambar 17.



Gambar 16 CRC cards iterasi ketiga



Gambar 17 Class diagram iterasi ketiga

Pengkodean (Coding)

Iterasi ketiga membuat API untuk modul lokasi menggunakan data lokasi seluruh Indonesia dalam format SQL. Data format SQL kemudian diubah menjadi format JSON untuk masing-masing tabel menggunakan phpMyAdmin. Data dalam format JSON kemudian dimasukkan ke dalam basis data mongoDB dengan menggunakan *terminal* untuk masing-masing tabel menjadi *collections* yang bersesuaian. *Code* di *terminal* yang digunakan untuk memasukkan data format JSON ke dalam basis data mongoDB dapat dilihat pada Gambar 18.

```

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

```

Gambar 18 Code memasukkan data format JSON ke basis data mongoDB

Pair programming dilakukan antara fungsi `getMailVerify` dengan fungsi `register` yang dapat dilihat pada Gambar 19. 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.isValidated, user.email, user.username, user.name, user.user_id);
res.json({"status": "200", "message": "Create User Success", data: user, token: token});

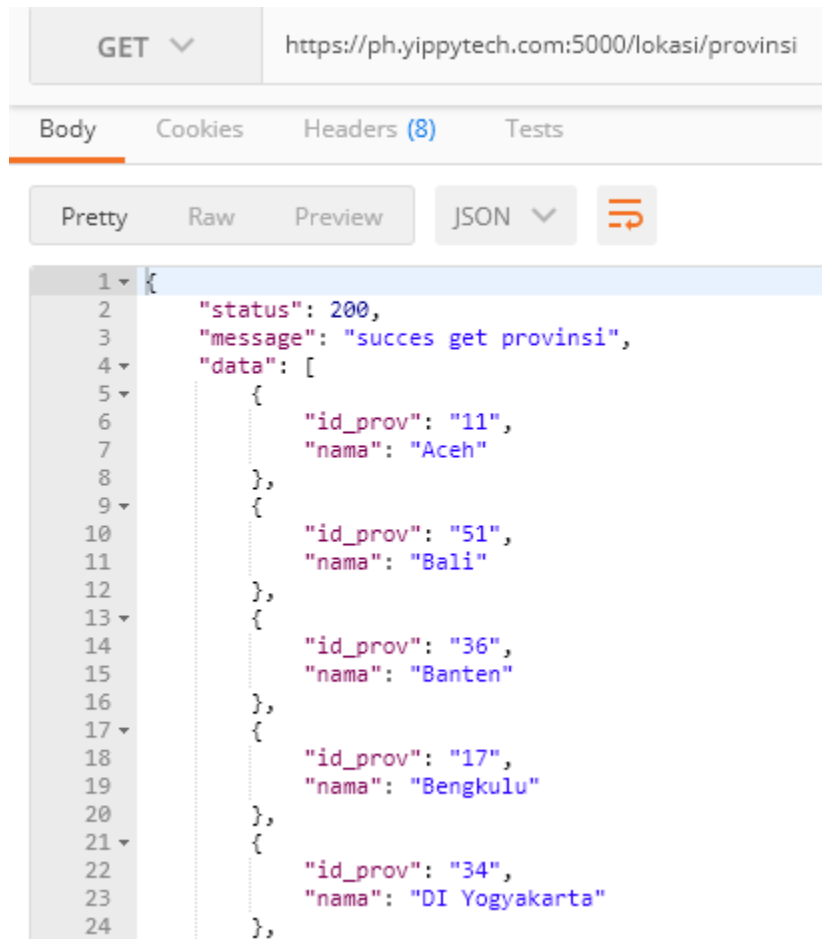
```

Gambar 19 Integrasi fungsi `getMailVerify` dengan fungsi `register`

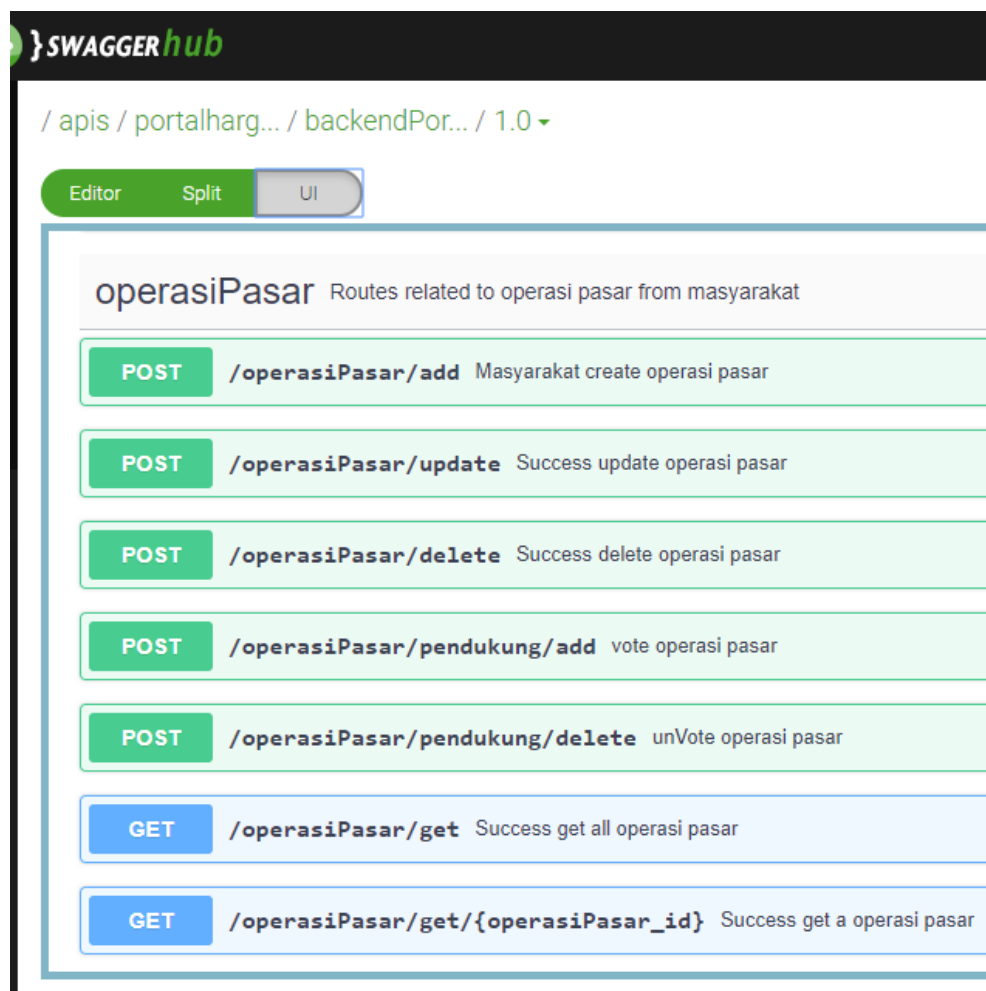
Pengujian (*Testing*)

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 8. Detail pengujian iterasi ketiga dapat dilihat pada Tabel 4. API yang sudah dikembangkan sudah didokumentasikan di <https://app.swaggerhub.com/apis/portalharga/backendPortalHarga/1.0>. Potongan dokumentasi API yang dapat dilihat pada Gambar 20.

Salah satu pengujian pada modul lokasi, yaitu pengujian fungsi melihat provinsi di Indonesia yang dapat dilihat pada Gambar 21. Pengujian fungsi melihat provinsi di Indonesia dengan mengakses URL <https://ph.yippytech.com:5000/lokasi/provinsi> menggunakan *method* GET. Format keluaran berupa JSON yang terdiri atas status, *message*, data, dan token. Data berisi informasi provinsi yang masing-masing terdiri dari *id_prov* dan nama provinsi.



Gambar 20 Pengujian fungsi melihat provinsi di Indonesia



Gambar 21 Dokumentasi API di Swaggerhub

Tabel 4 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
Mail	GET	Mengirim email verifikasi	/user/email/getMailVerify	Berhasil
	POST	Mengirim ulang email verifikasi	/user/email/validate/resend	Berhasil

<i>Class</i>	<i>Method</i>	Fungsi	URL	Status
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

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 pengkodean perlu dianalisis efisiensi algoritma 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, memaksimalkan 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.
- Ashana LN. 2015. Pertukaran data antara Rstudio dan MongoDB pada sistem informasi geografis untuk kasus pertanian Indonesia [skripsi]. Bogor (ID): Institut Pertanian Bogor.
- [APJII] Asosiasi Penyelenggara Jasa Internet Indonesia. 2014. Profil pengguna internet Indonesia 2014 [Internet]. [diunduh 2016 Des 22]. Tersedia pada: <http://www.slideshare.net/internetsehat/profil-pengguna-internet-indonesia-2014-riset-oleh-apjii-dan-puskakom-ui>.
- Cohn M. 2014. User Stories [Internet]. [diunduh 2017 Jan 16]. Tersedia pada: <https://www.mountangoatsoftware.com/agile/user-stories>.
- Elian N, Lubis DP, Rangkuti PA. 2014. Penggunaan internet dan pemanfaatan informasi pertanian oleh penyuluh pertanian di Kabupaten Bogor wilayah Barat. *Jurnal Komunikasi Pembangunan*. 12(2):105–106.
- Haviluddin, 2011. Memahami penggunaan UML (*Unified Modeling Language*). *Jurnal Informatika Mulawarman*, 6(1): 5–6.
- Herdiana Y. 2014. Aplikasi rumus matematika SMA berbasis mobile. *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)*. 1(1), ISSN: 2089–9033.
- [Kementan] Kementerian Pertanian. 2015. Renstra kementan tahun 2015-2019 [Internet]. [diunduh 2017 Mei 13]. Tersedia pada: http://www.pertanian.go.id/file/RENSTRA_2015-2019.pdf
- Kurniawan E. 2014. Implementasi REST Web service untuk *sales order* dan *sales tracking* berbasis *mobile*. *Jurnal EKSIS*. 7(1): 1–12.
- Oktaviani N, Hutrianto. 2016. *Extreme programming* sebagai metode pengembangan e-keuangan pada pondok pesantren qodratullah. *Jurnal Ilmiah MATRIK*. 18(2):165–168.
- Pressman RS. 2010. *Software Engineering: A Practitioner's Approach Ed ke-7*. Boston (US): Mc Graw Hill.
- Rismanto R, Arhandi PP, Prasetyo A. 2016. Rancang bangun aplikasi ujian *online real time* dengan menggunakan arsitektur *mean*. *Jurnal Teknologi Informasi*. 7(2): 150–151.
- Rohman FN. 2015. Pengembangan aplikasi web pengolah data nilai lomba baris berbaris menggunakan metodologi *Extreme Programming* [skripsi]. Yogyakarta (ID): Universitas Negeri Yogyakarta.
- Saputra ZR. 2015. Aplikasi SMS *Center* untuk informasi harga komoditi hasil pertanian Kabupaten Ogan Ilir. *Jurnal SIGMATA*. 4(2): 3–6.
- Setiady H. 2013. *Sistem informasi pemesanan dan penjualan berbasis web pada dewi florist* [skripsi]. Palembang (ID): Sekolah Tinggi Ilmu Manajemen Informatika dan Komputer *Global Information Multi Data* Palembang.
- Setiana D. 2016. Pengembangan modul pelaporan harga komoditas pertanian pada sisi pengguna admin dan pemerintah menggunakan REST API [skripsi]. Bogor (ID): Institut Pertanian Bogor.

- Setyatama D. 2016. Pengembangan aplikasi pelaporan harga komoditas pertanian berbasis *mobile* menggunakan REST API [skripsi]. Bogor (ID): Institut Pertanian Bogor.
- Undang-Undang Republik Indonesia Nomor 8 Tahun 1999 Tentang Perlindungan Konsumen. [Internet]. [diunduh 2017 Jan 02]. Tersedia pada: https://www.kontras.org/uu_ri_ham/UU%20Nomor%208%20Tahun%201999%20tentang%20Perlindungan%20Konsumen.pdf
- Wihono A. 2009. Analisis volatilitas harga sayuran di Pasar Induk Kramat Jati [skripsi]. Bogor (ID): Institut Pertanian Bogor.

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 perdagangkan

Lampiran 2 Daftar *collections* iterasi pertamaa *Collections* laporan harga

```

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});
--

```

b *Collections* operasi pasar

```

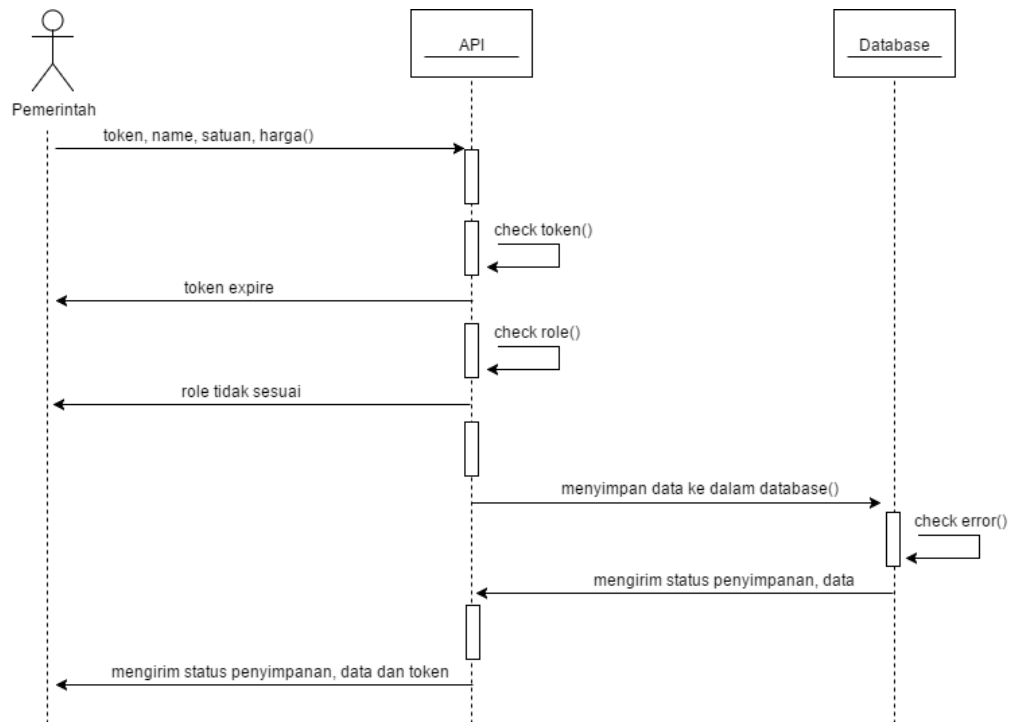
9  var operasiPasarModel = new Schema({
10     user_id:Number,
11     komoditas_id:Number,
12     latitude:Number,
13     longitude:Number,
14     alamat:String,
15     pesan:String,
16     datePost:Number,
17     //array pendukungnya
18     pendukung:[{
19         user_id:Number
20     }],
21     tanggapan:[
22         {
23             user_id:String,
24             isi:String,
25             datePost:Number
26         }
27     ]
28 });
29
30 operasiPasarModel.plugin(autoIncrement.plugin,{model:'operasiPasar',
31     field:'operasiPasar_id',startAt:1});
--

```

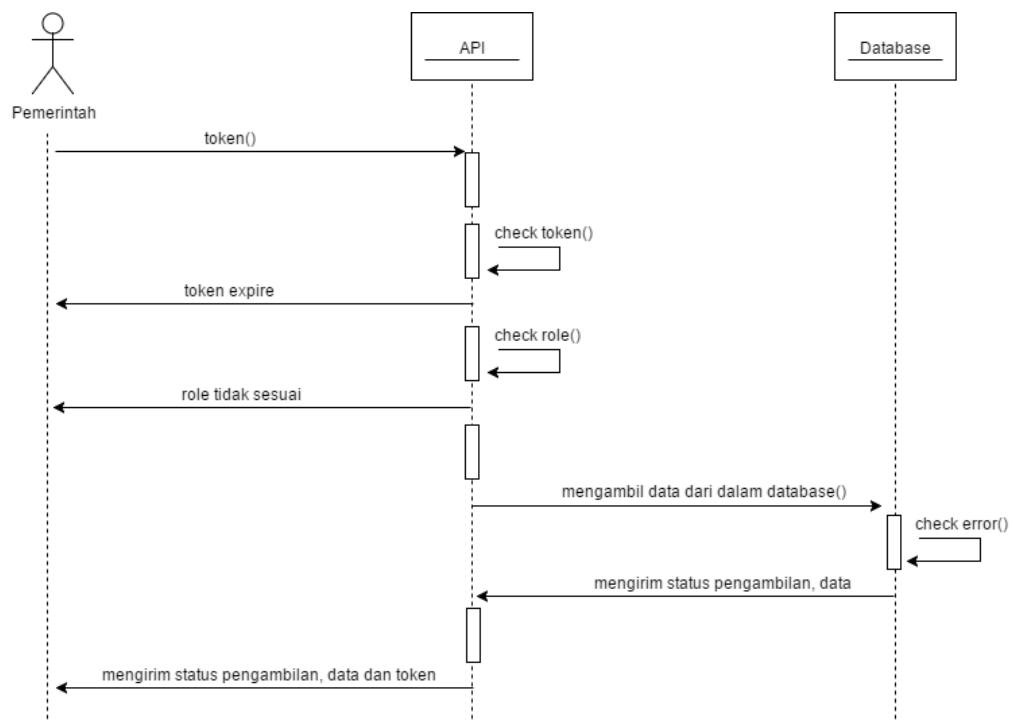
Lampiran 3 *Sequence diagram* iterasi pertama

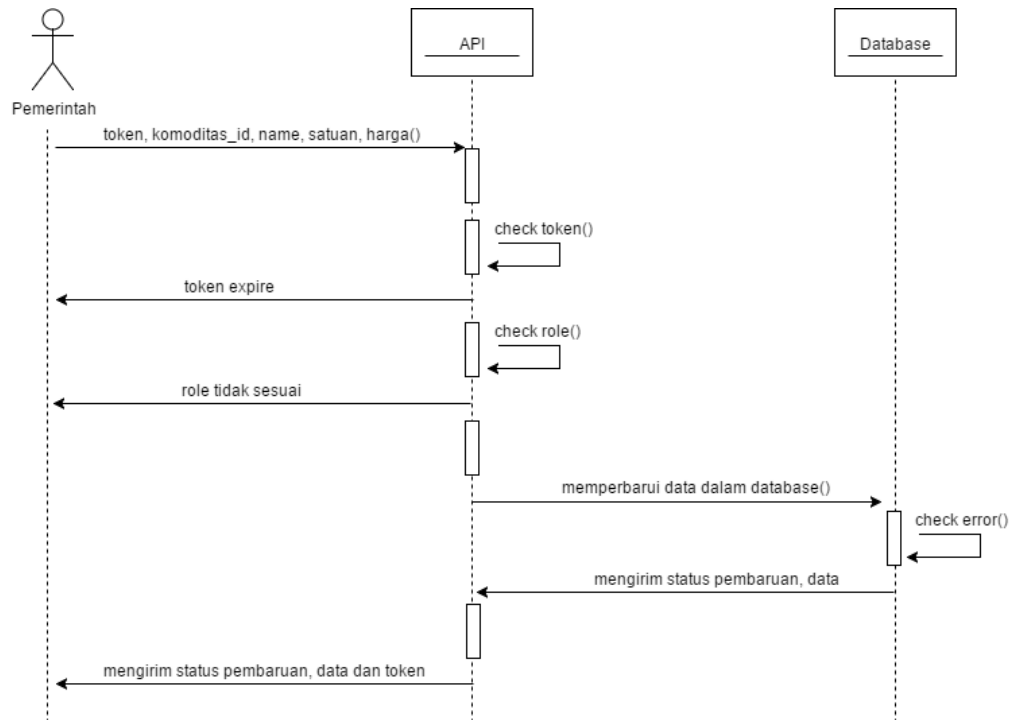
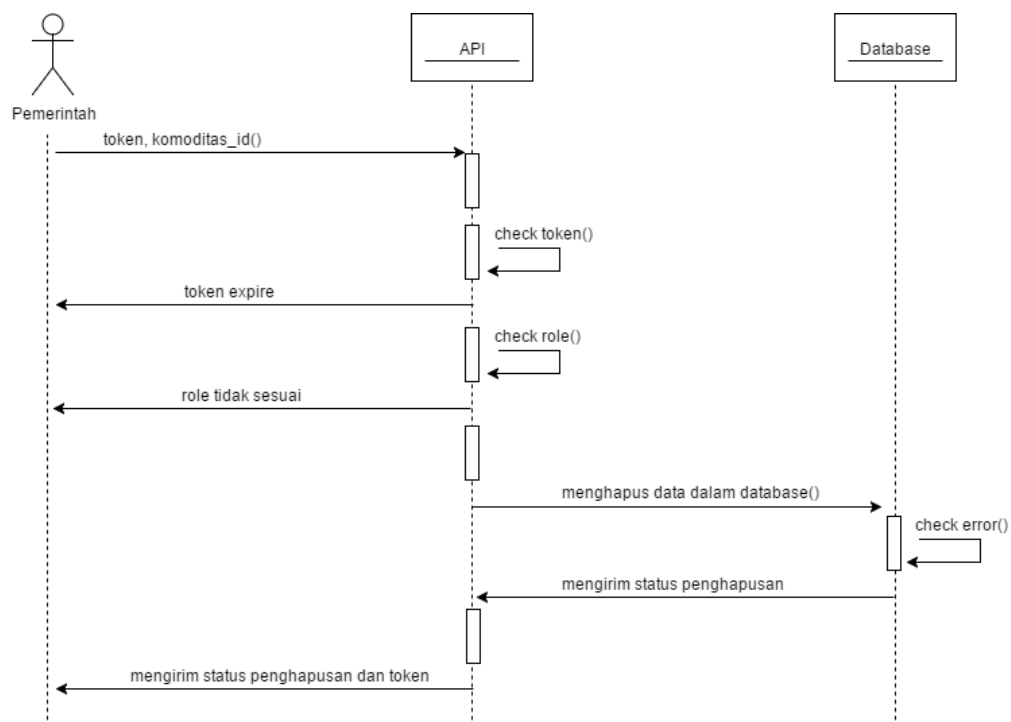
Komoditas

a *Create*



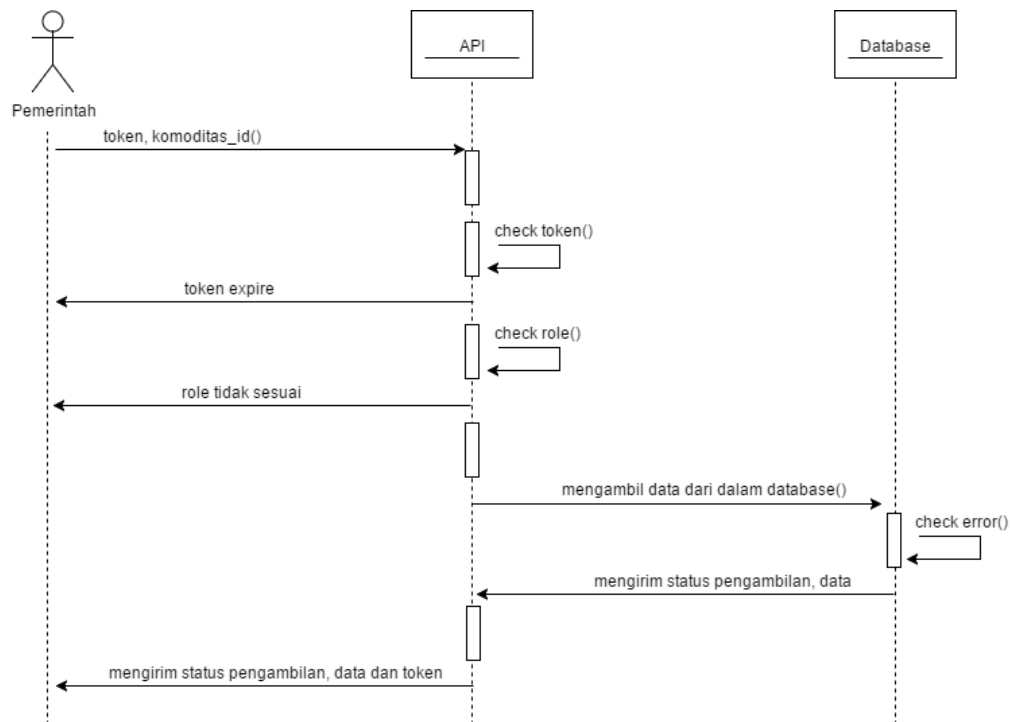
b *Read*



Lampiran 3 *Sequence diagram* iterasi pertama (lanjutan)c *Update*d *Delete*

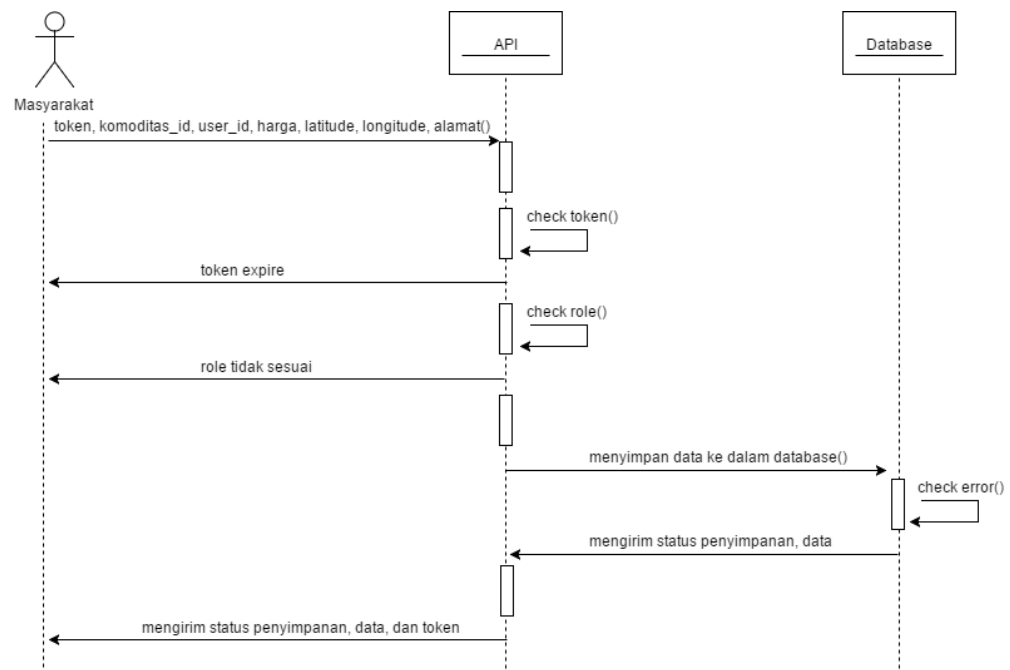
Lampiran 3 *Sequence diagram* iterasi pertama (lanjutan)

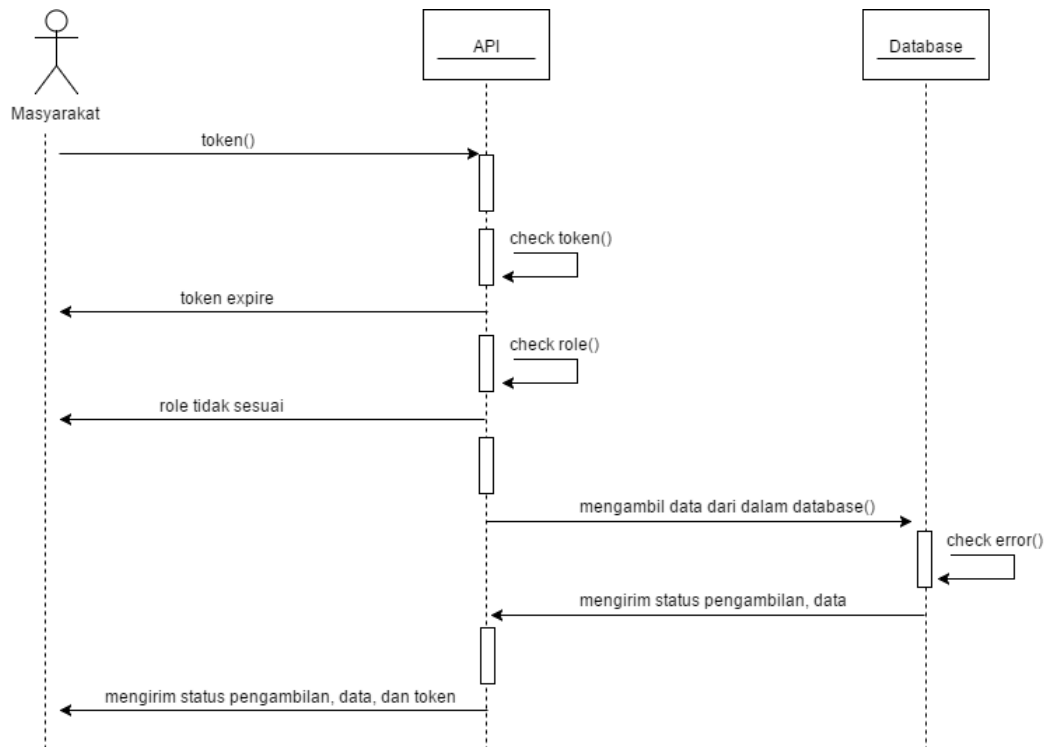
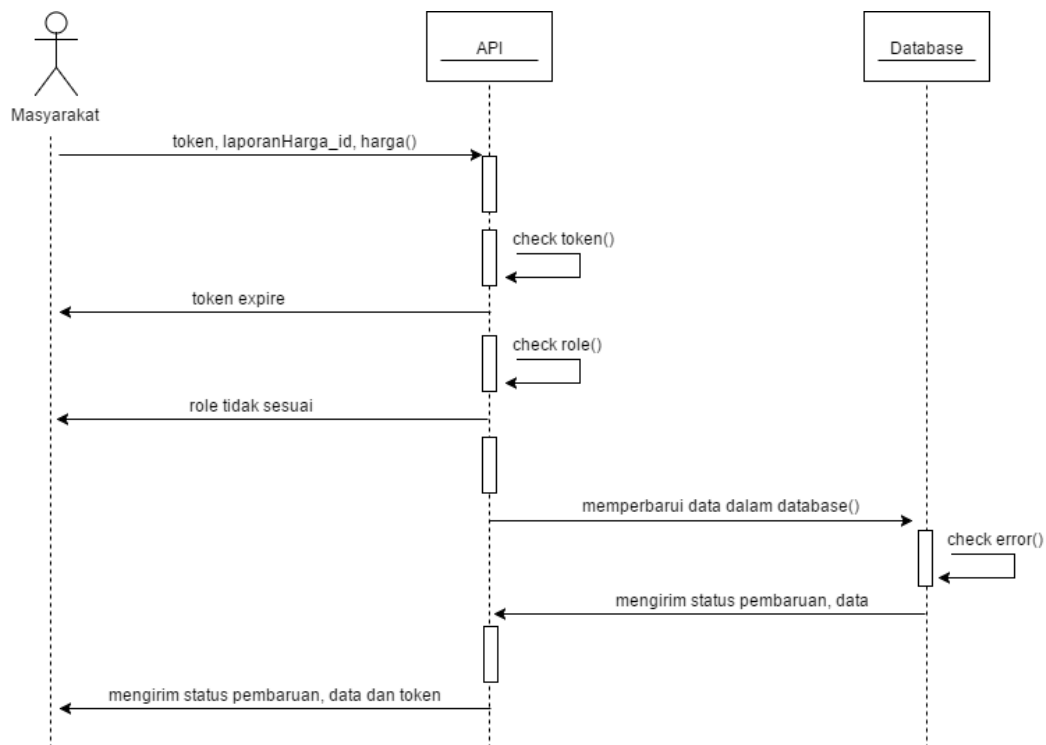
e *Read one*



Laporan harga

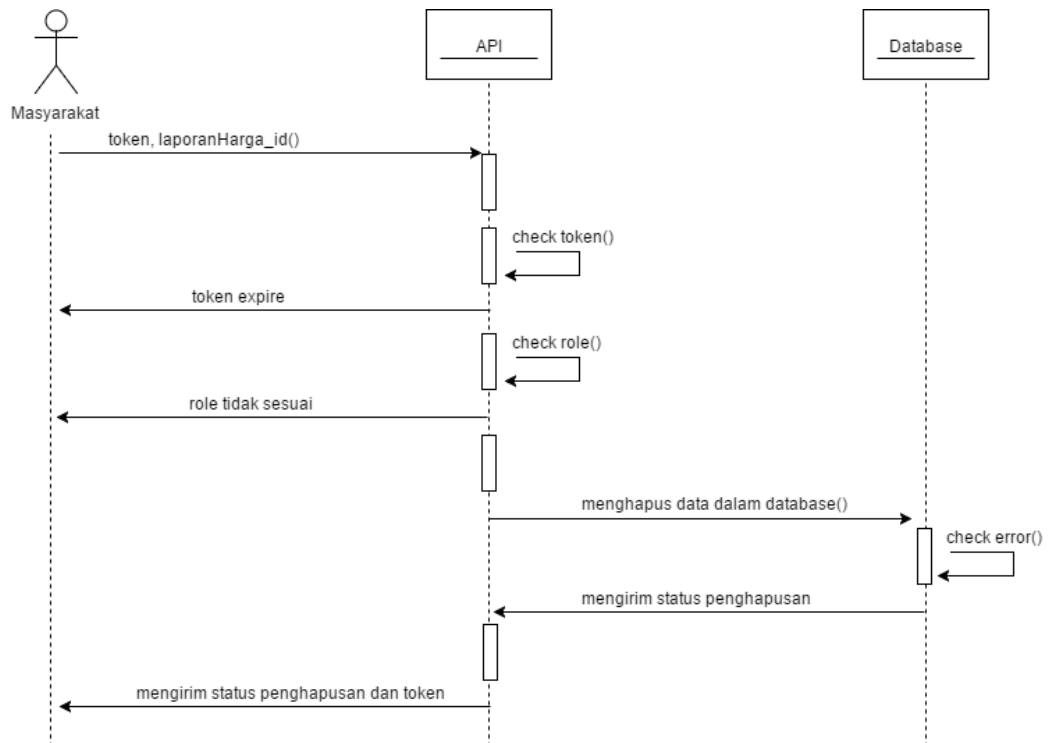
a *Create*



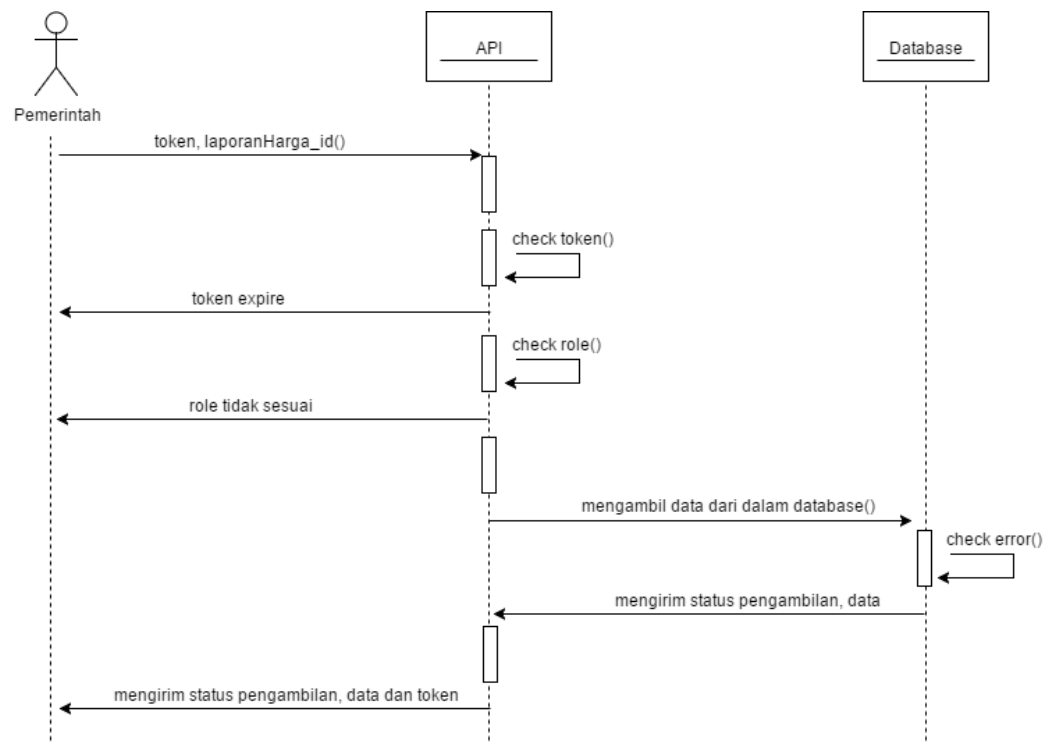
Lampiran 3 *Sequence diagram* iterasi pertama (lanjutan)b *Read*c *Update*

Lampiran 3 *Sequence diagram* iterasi pertama (lanjutan)

d *Delete*

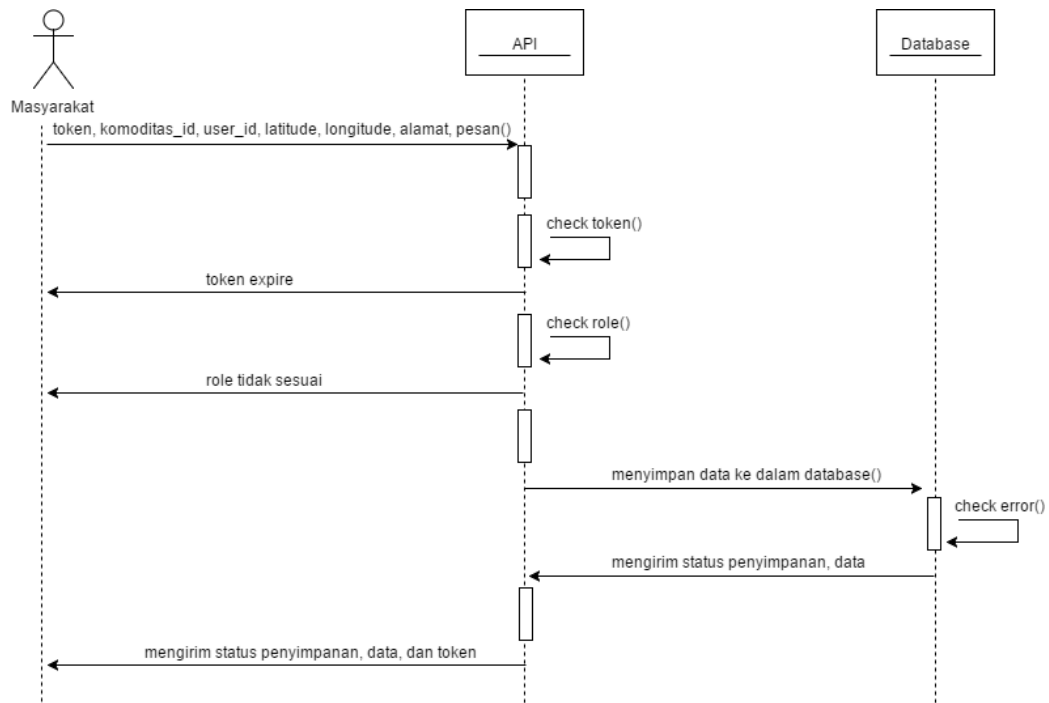
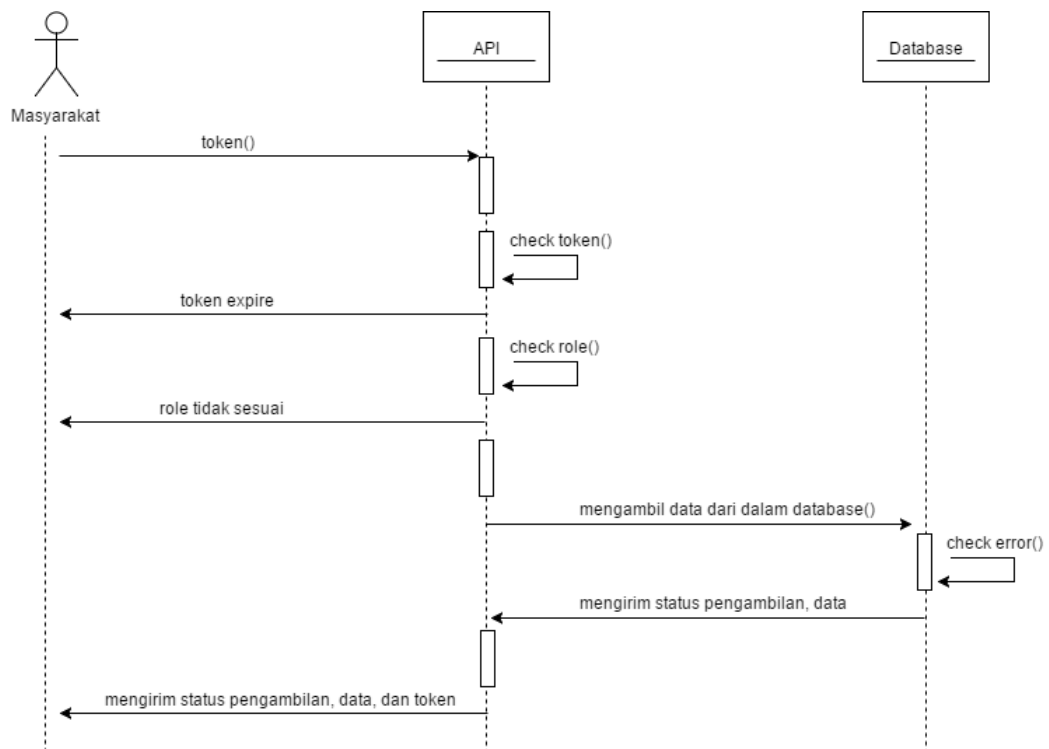


e *Read one*



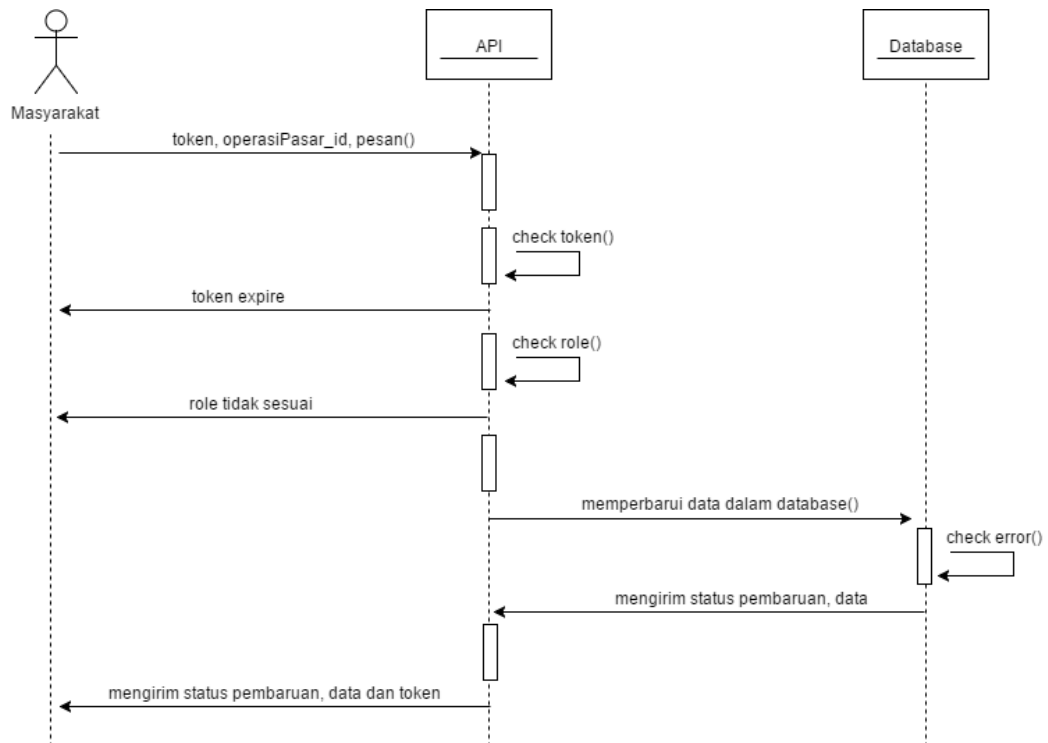
Lampiran 3 *Sequence diagram* iterasi pertama (lanjutan)

Operasi pasar

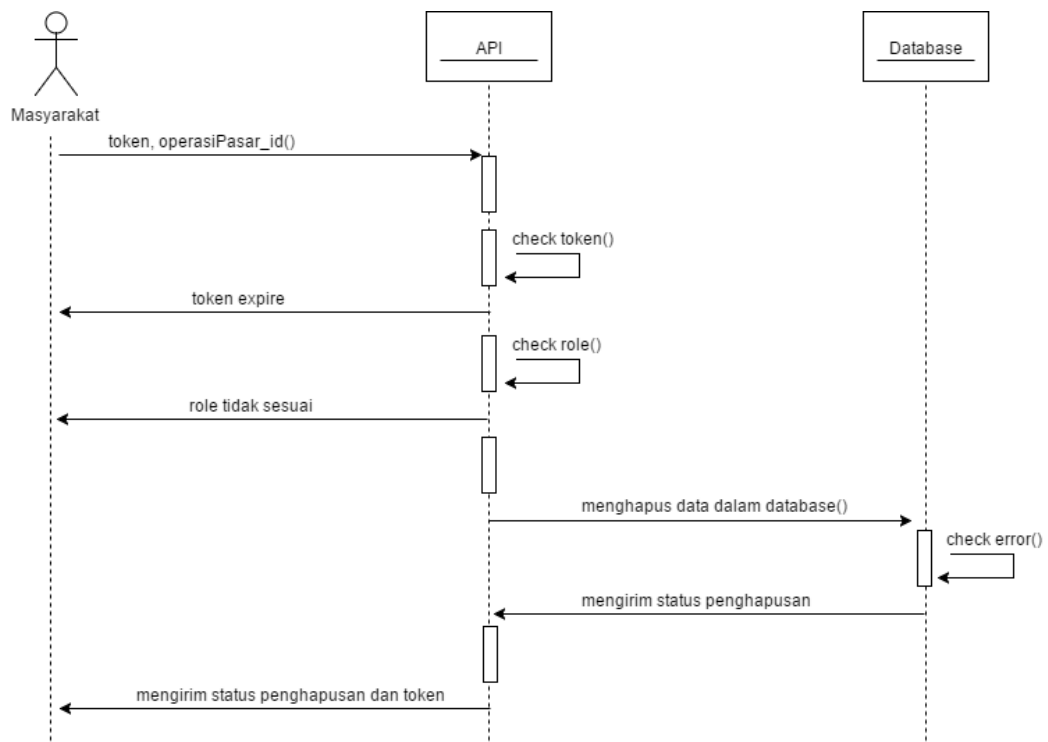
a Create*b Read*

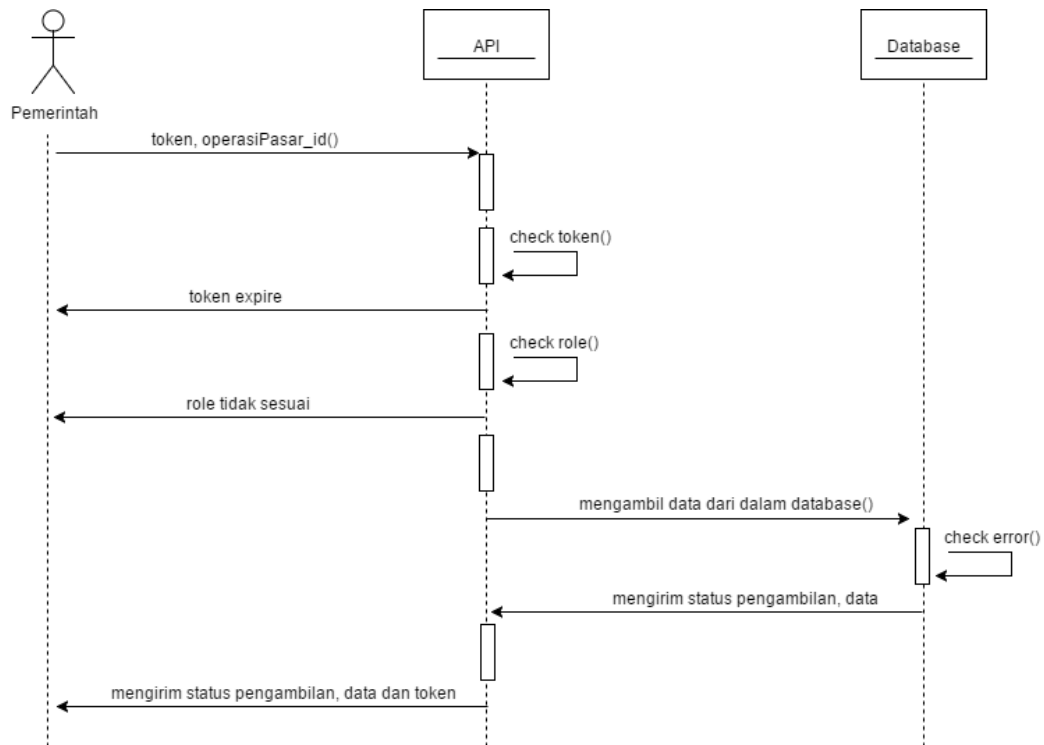
Lampiran 3 *Sequence diagram* iterasi pertama (lanjutan)

c *Update*



d *Delete*



Lampiran 3 *Sequence diagram* iterasi pertama (lanjutan)*e Read one*

Lampiran 4 *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 detail 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 4 *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 	<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

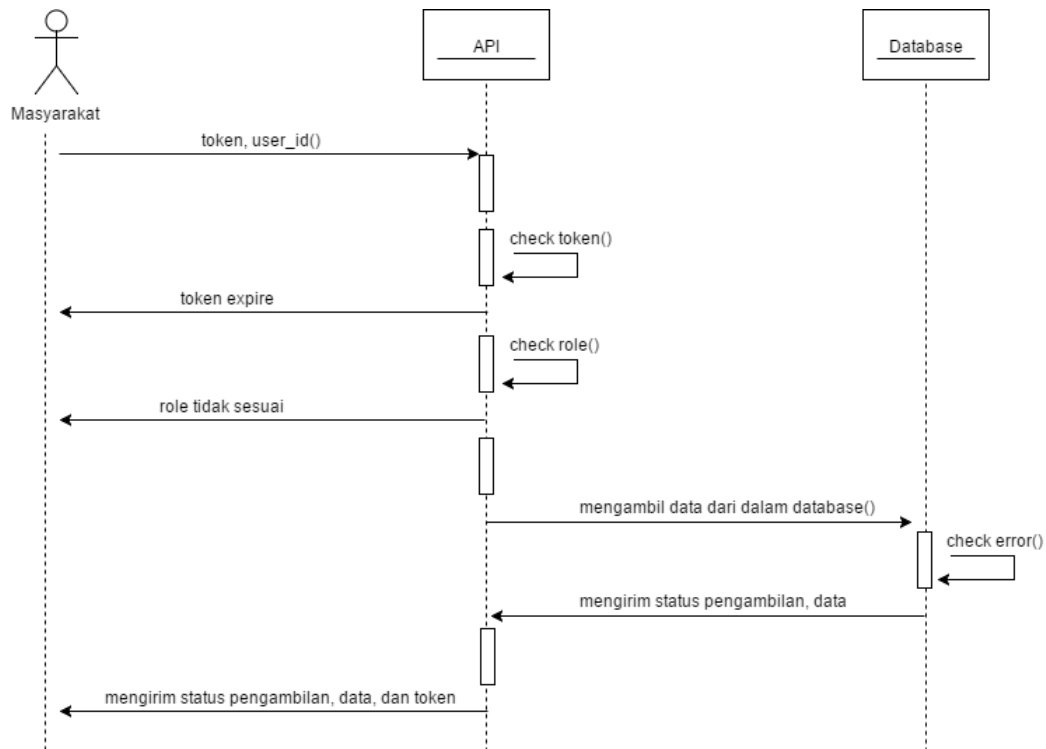
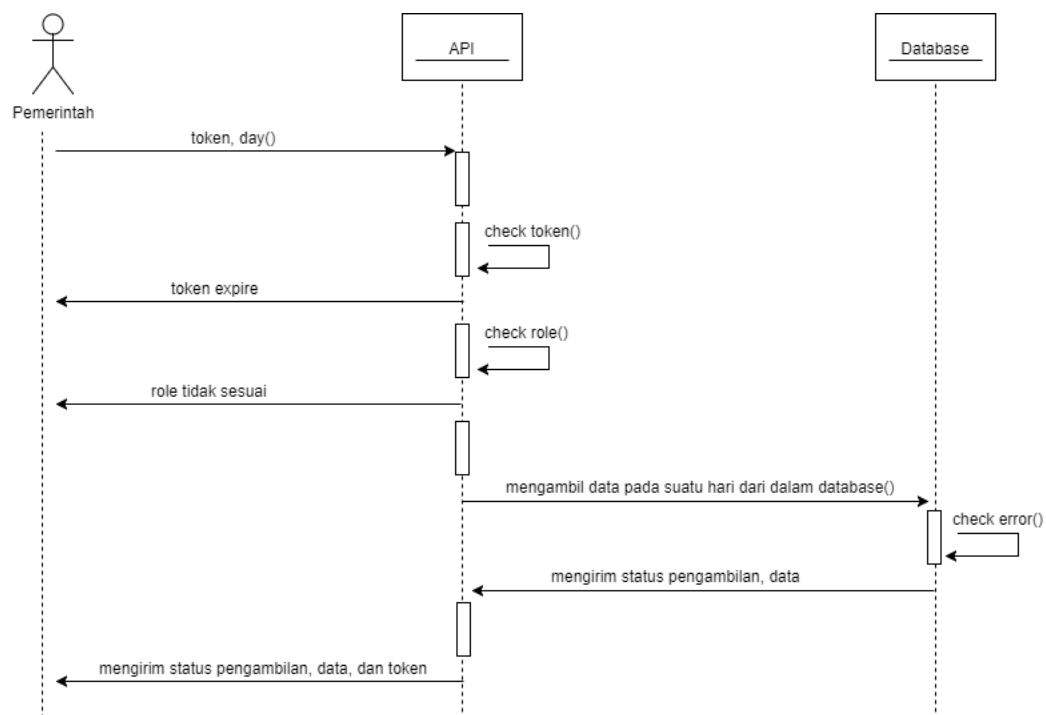
Lampiran 4 *Acceptance test criteria* iterasi pertama (lanjutan)

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 5 *Sequence diagram* iterasi kedua

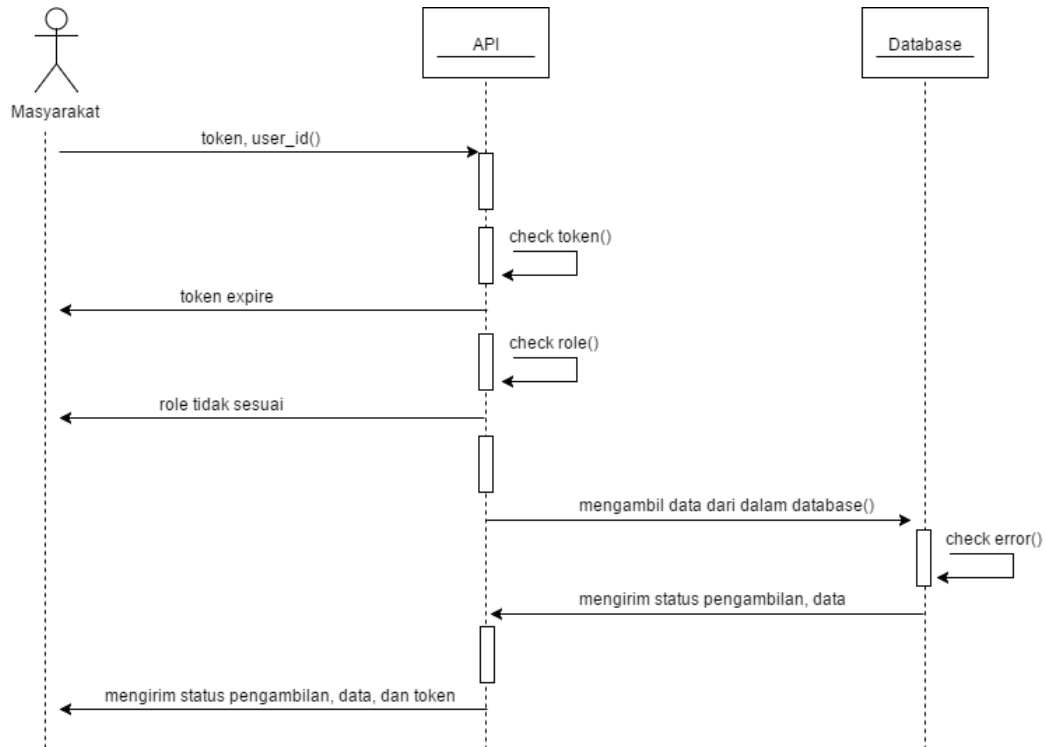
Laporan Harga

a *History* penggunab *Day* laporan

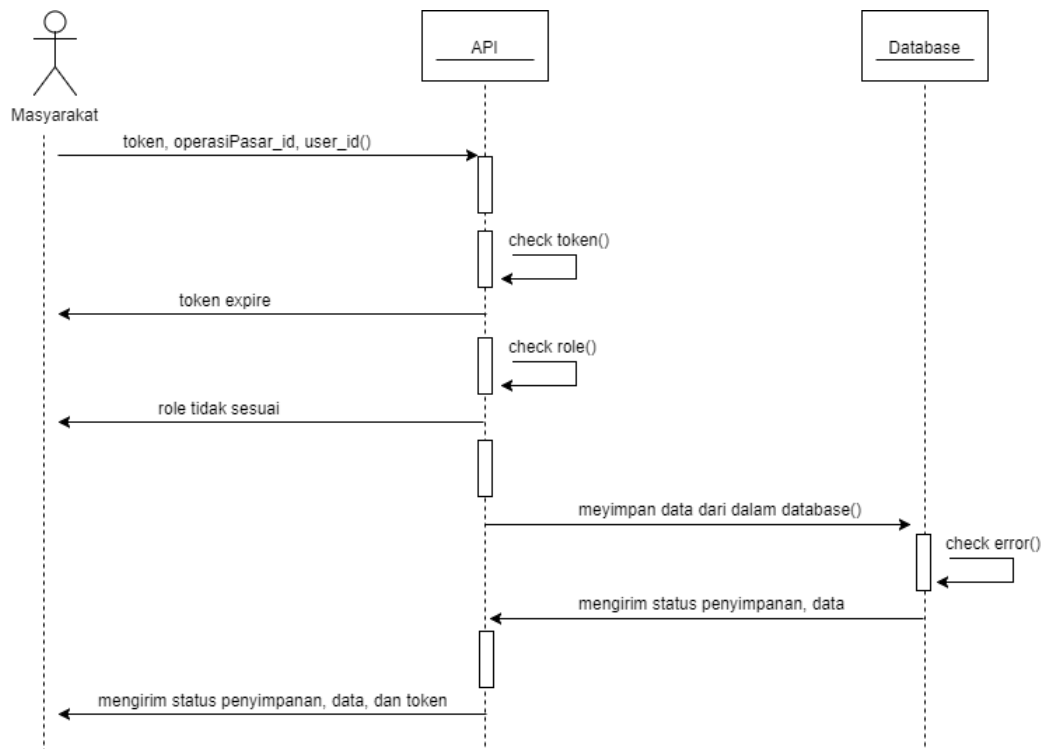
Lampiran 5 *Sequence diagram* iterasi kedua (lanjutan)

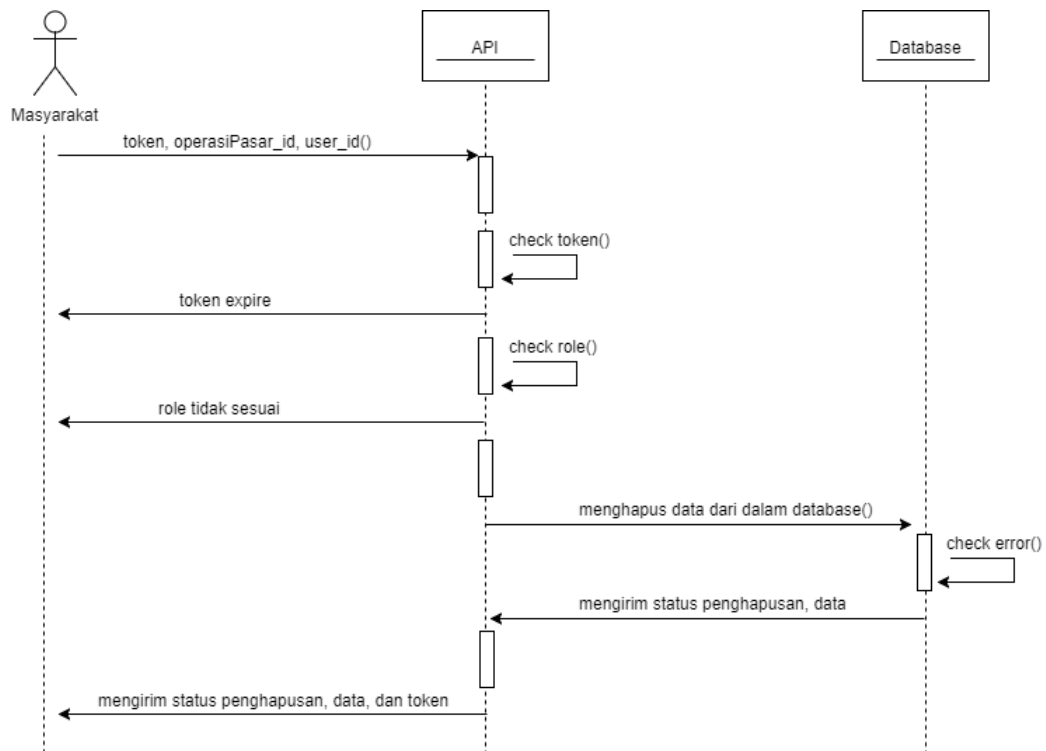
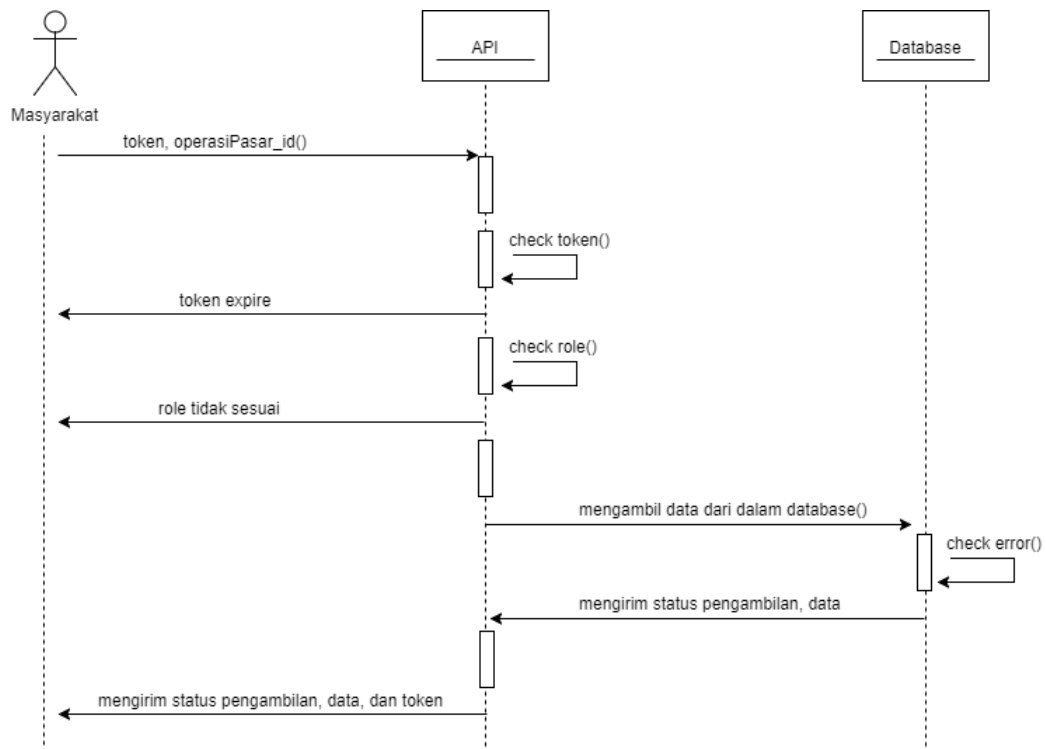
Operasi pasar

a *History* pengguna



b *Vote* operasi pasar



Lampiran 5 *Sequence diagram* iterasi kedua (lanjutan)c *UnVote* operasi pasard *Get* pendukung operasi pasar

Lampiran 6 *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 6 *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

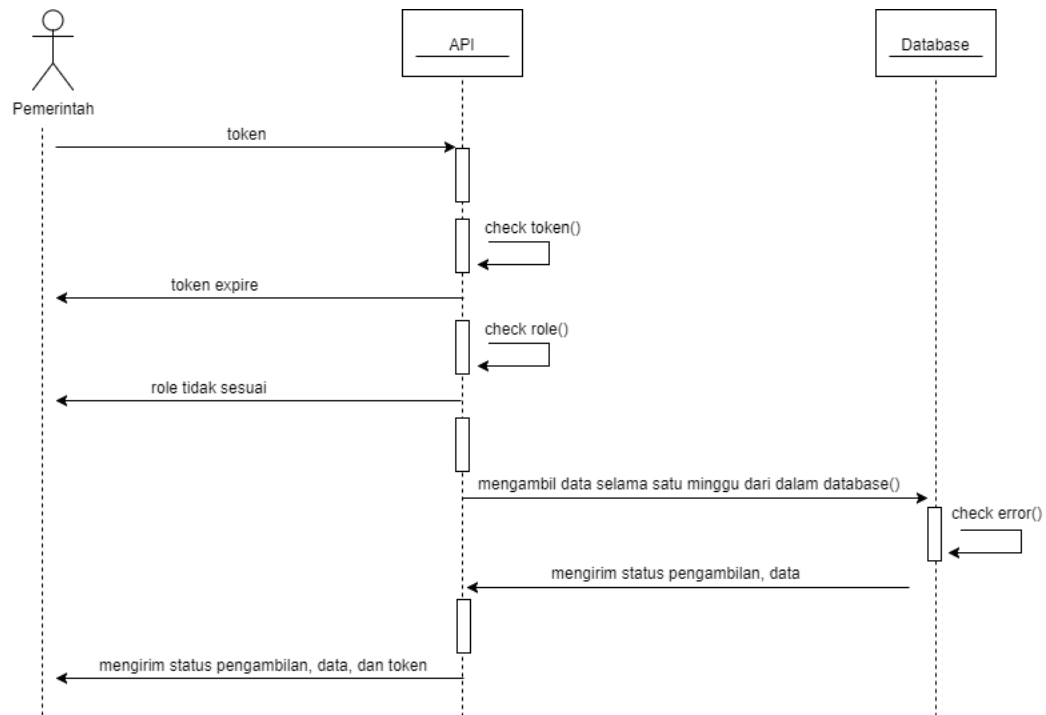
c Email

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

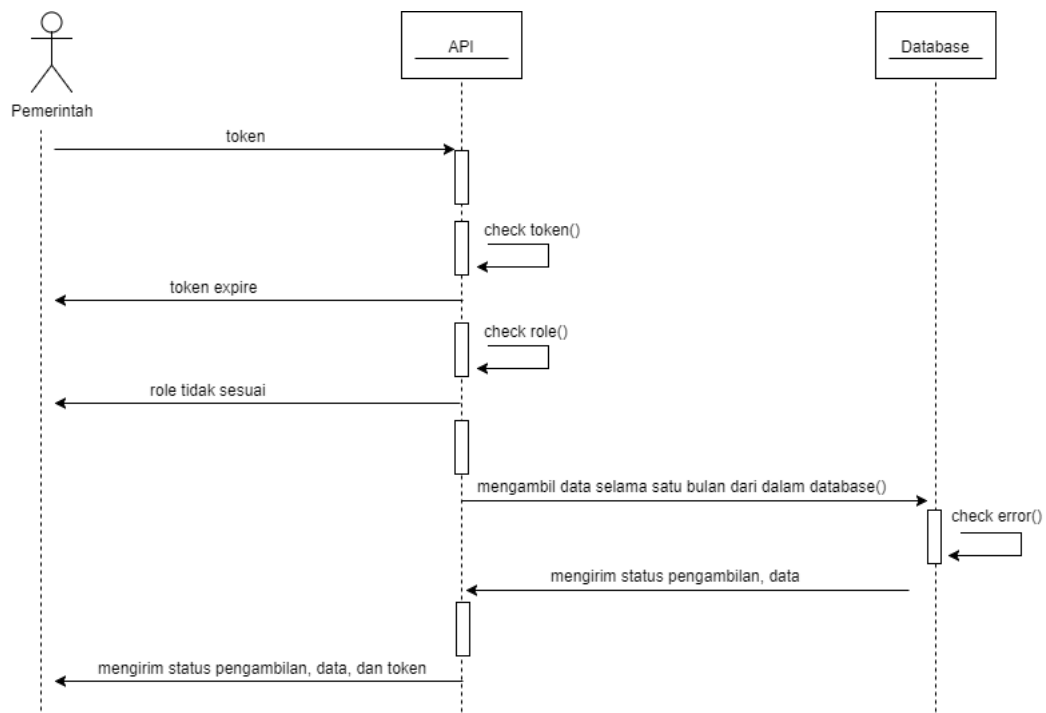
Lampiran 7 *Sequence diagram* iterasi ketiga

Laporan harga

a *Week* laporan harga

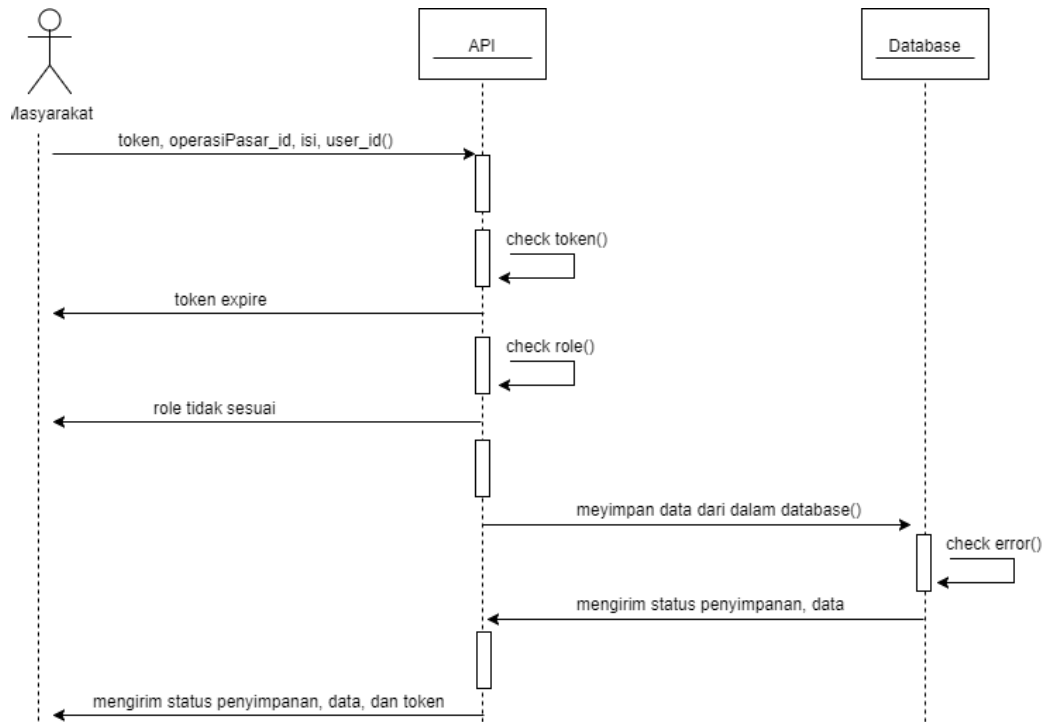
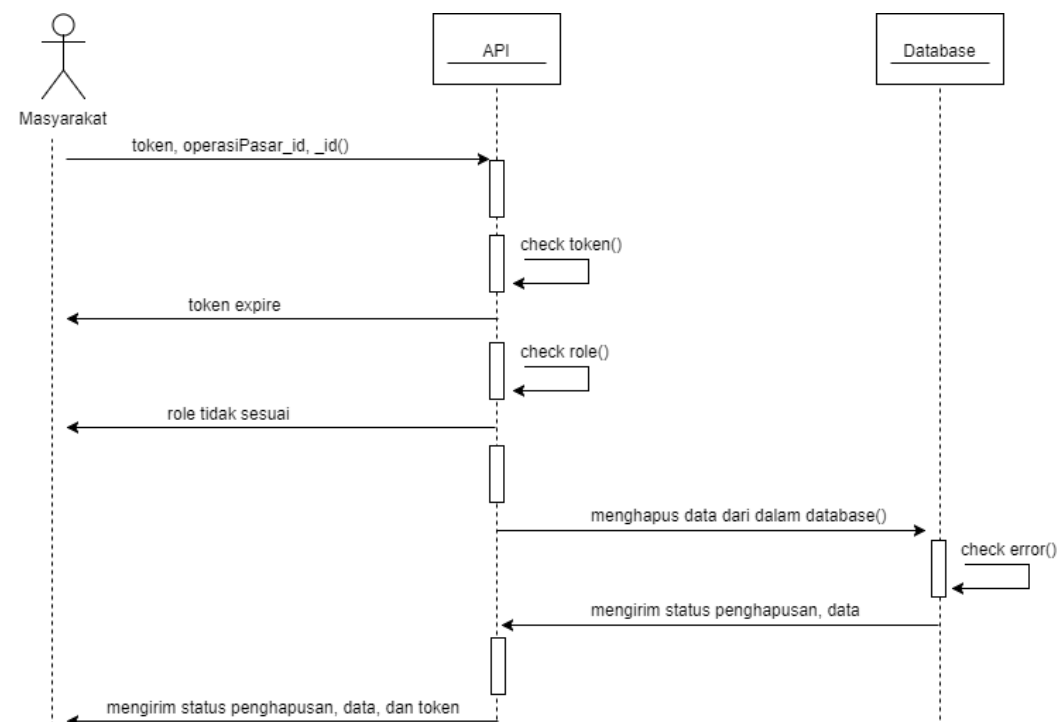


b *Month* laporan harga



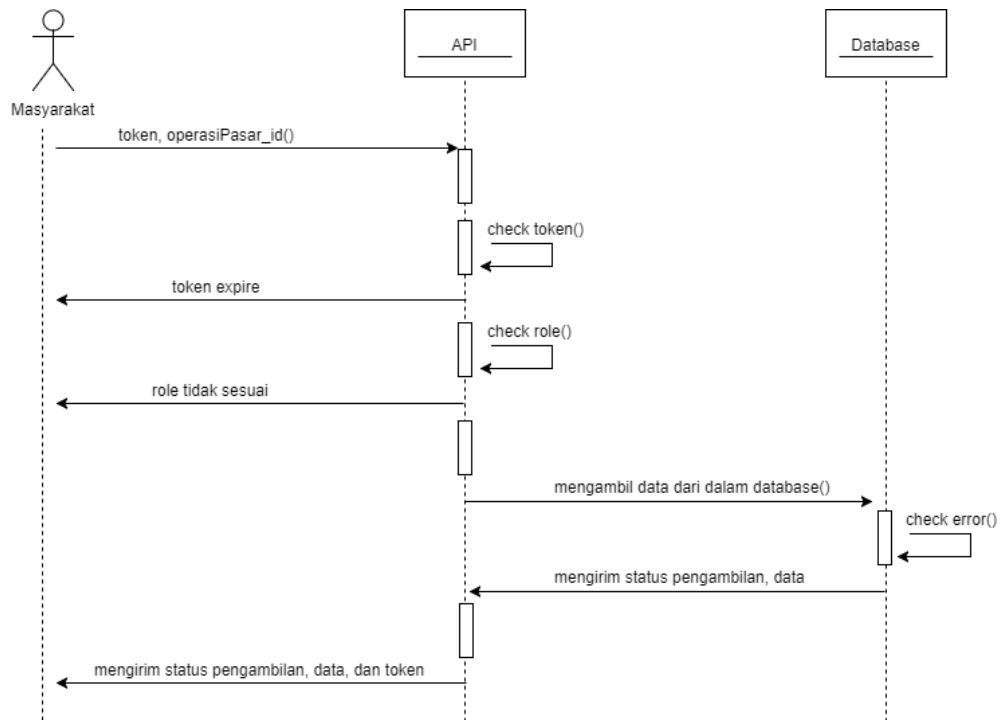
Lampiran 7 *Sequence diagram* iterasi ketiga (lanjutan)

Operasi pasar

a *Add* tanggapan operasi pasarb *Delete* tanggapan operasi pasar

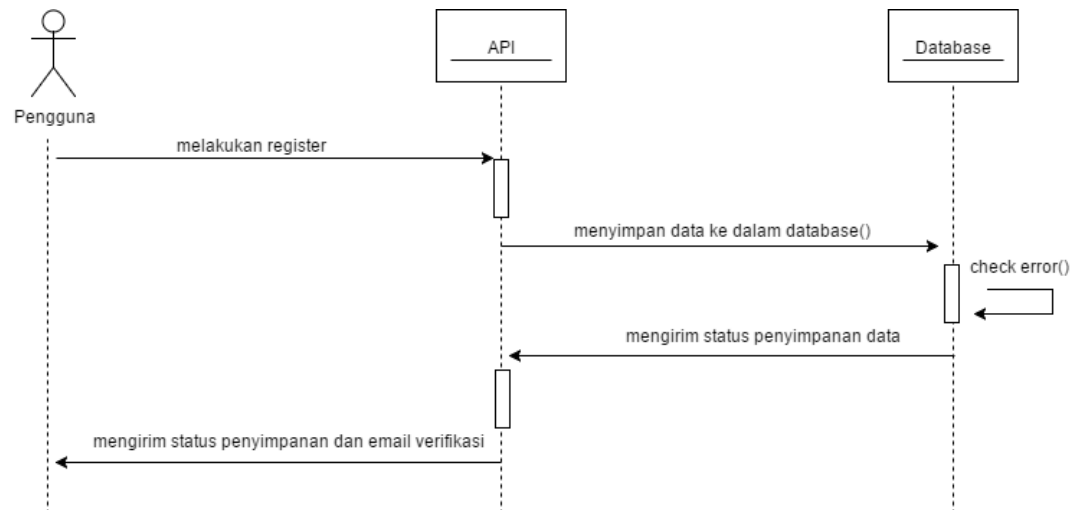
Lampiran 7 *Sequence diagram* iterasi ketiga (lanjutan)

c *Get* tanggapan operasi pasar



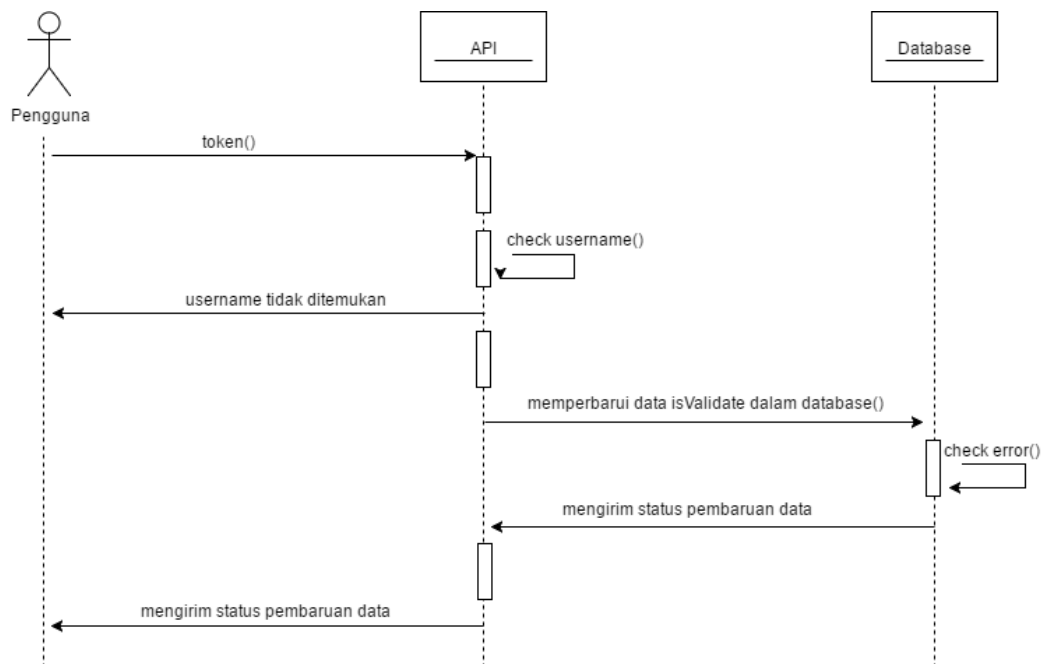
Email

a *getMailVerify*

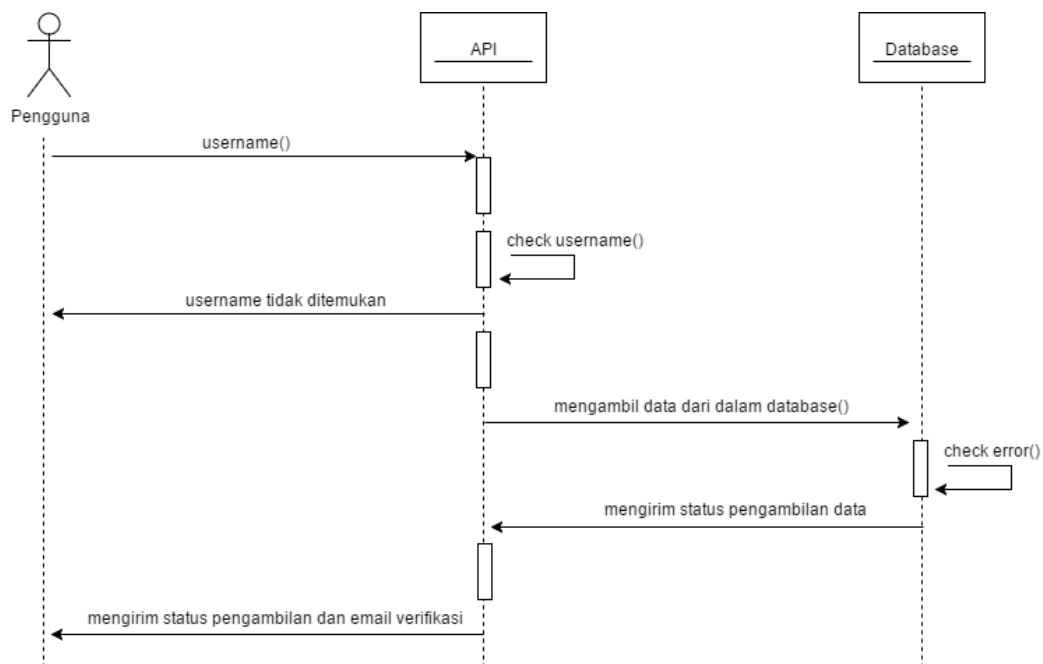


Lampiran 7 *Sequence diagram* iterasi ketiga (lanjutan)

b postVerify

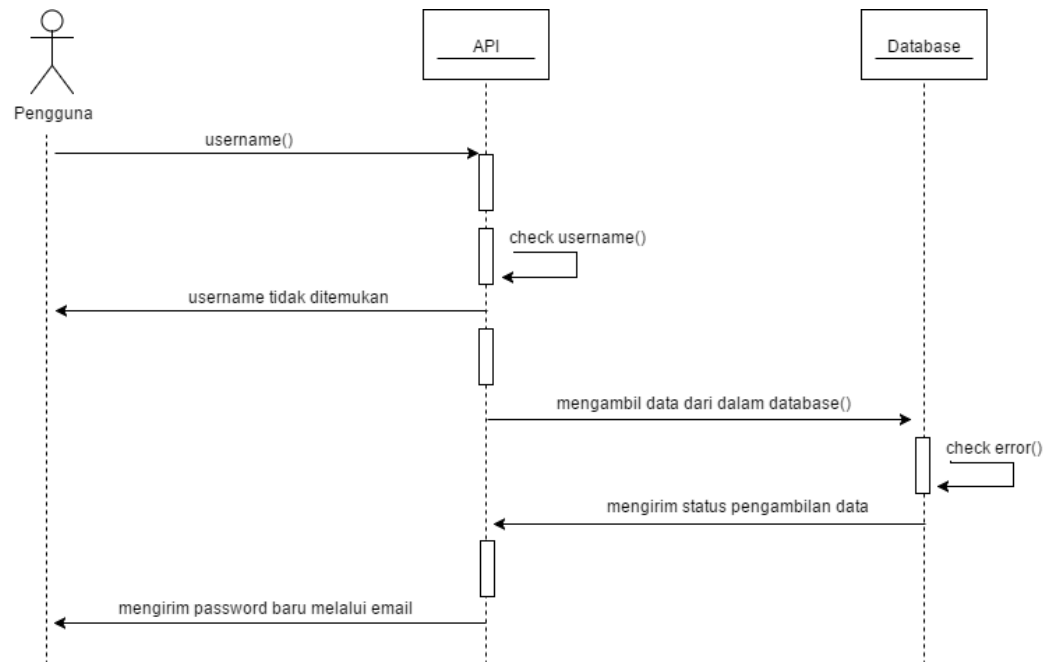


c reSendMailVerify

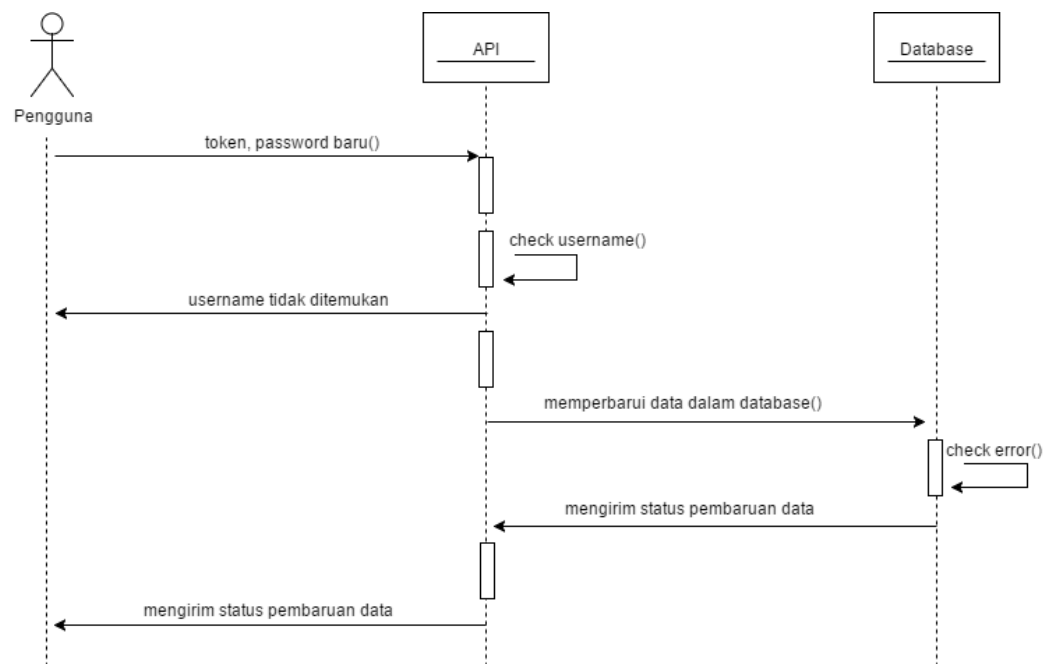


Lampiran 7 *Sequence diagram* iterasi ketiga (lanjutan)

d forgetPassword



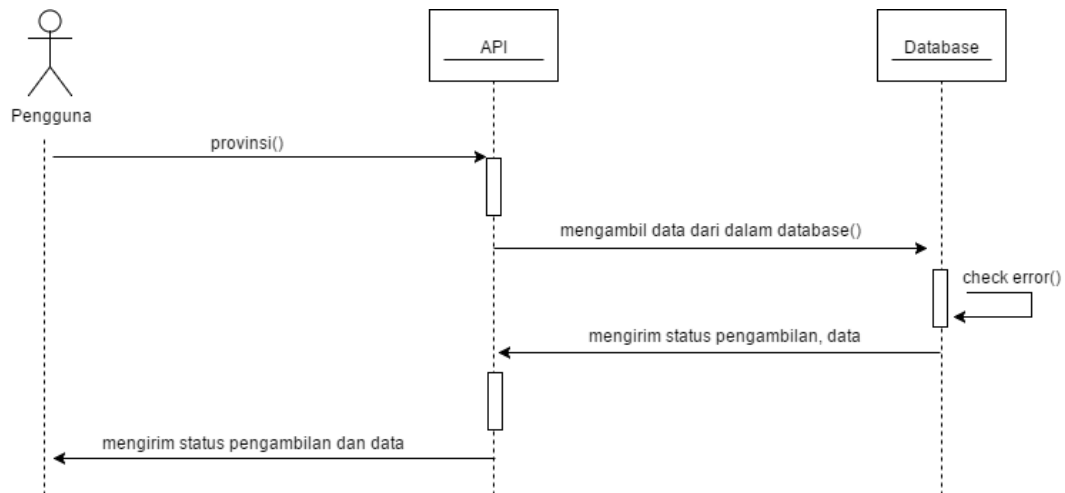
e reNewPassword



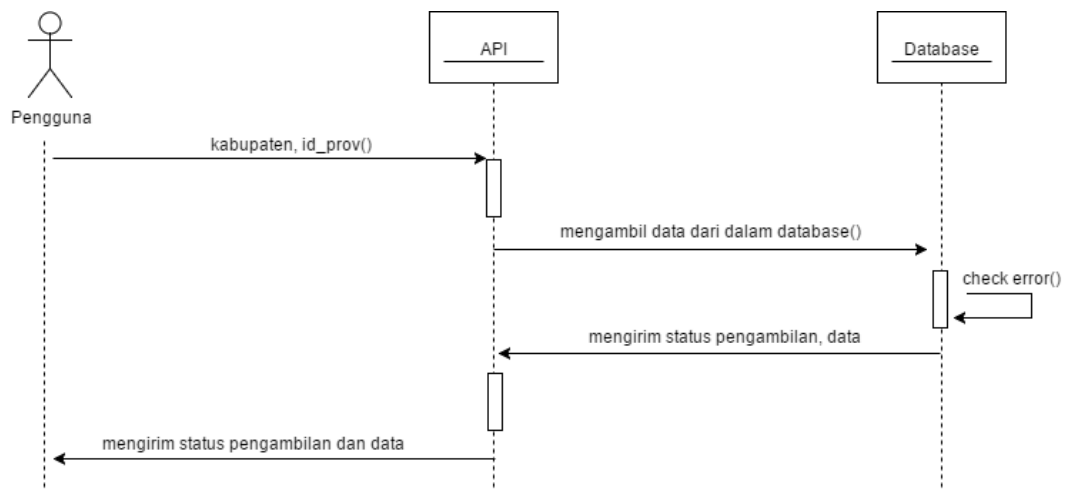
Lampiran 7 *Sequence diagram* iterasi ketiga (lanjutan)

Lokasi

a Provinsi

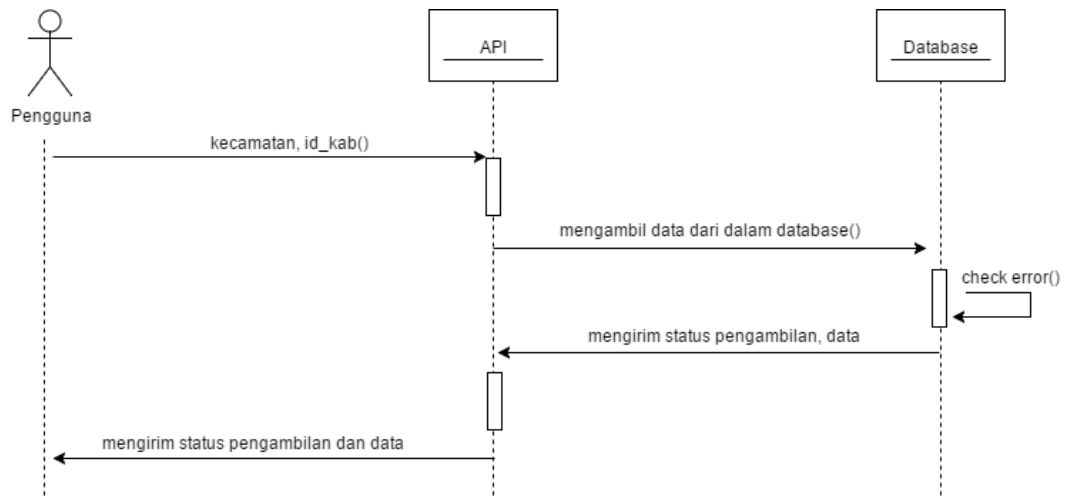


b Kabupaten

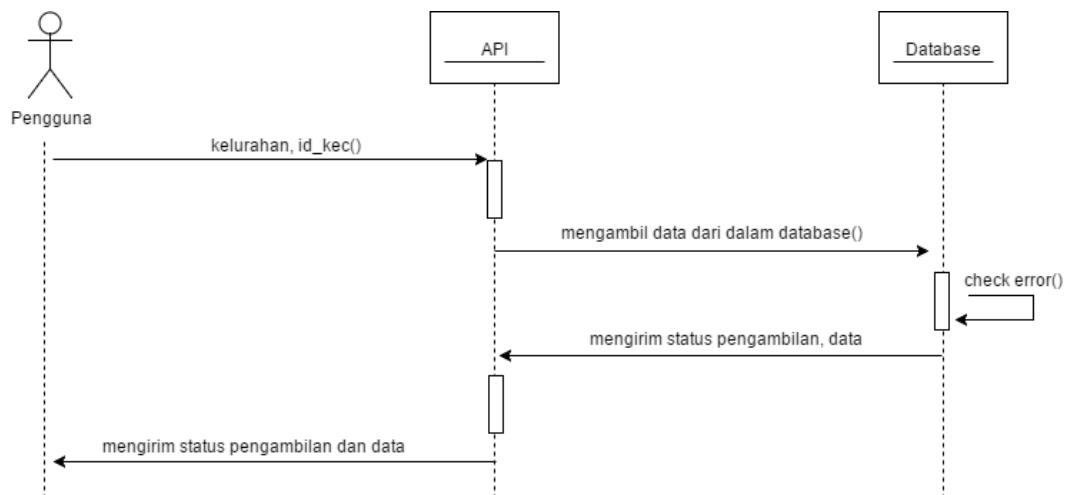


Lampiran 7 *Sequence diagram* iterasi ketiga (lanjutan)

c Kecamatan



d Kelurahan



Lampiran 8 *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 8 *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 8 *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.