

**PENGEMBANGAN MODUL PELAPORAN APLIKASI AKTIVITAS KAMPANYE UNTUK MASYARAKAT DAN PEMERINTAH MENGGUNAKAN REST APIS**

**ILMU KOMPUTER**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**INSTITUT PERTANIAN BOGOR**

**BOGOR**

**2016**

**INDRIYANI**

**PERNYATAAN MENGENAI SKRIPSI DAN  
SUMBER INFORMASI SERTA PELIMPAHAN HAK CIPTA**

Dengan ini saya menyatakan bahwa skripsi berjudul Pengembangan Modul Pelaporan Aplikasi Aktivitas Kampanye untuk Masyarakat dan Pemerintah Menggunakan REST APIs 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 2016

*Indriyani*

NIM G64120028

**ABSTRAK**

INDRIYANI. Pengembangan Modul Pelaporan Aplikasi Aktivitas Kampanye untuk Masyarakat dan Pemerintah Menggunakan REST APIs. Dibimbing oleh DEAN APRIANA RAMADHAN.

Pengawasan aktivitas kampanye pada pemilihan umum (Pemilu) dilakukan oleh Badan Pengawas Pemilu (Bawaslu). Bawaslu memiliki kendala dalam pelaksanaan pengawasan yaitu kendala daya dukung pengawasan berupa keterbatasan jumlah Pengawas Pemilu Lapangan (PPL). Melihat kendala tersebut dibutuhkan media untuk pelaporan yang dapat dilakukan oleh masyarakat. Penelitian ini merupakan pengembangan modul pelaporan aplikasi aktivitas kampanye untuk masyarakat dan pemerintah menggunakan *REpresentational State Transfer (REST) Application Programming Interface* (API) dengan struktur basis data NoSQL. Metode yang digunakan dalam penelitian ini adalah *Extreme Programming.* Pemasukkan data pada REST API menggunakan Node.js dan hasil dari data tersebut berupa format teks JSON. Penelitian ini telah berhasil mengembangkan aplikasi *web* yang dapat membantu masyarakat dalam mengetahui informasi seputar calon legislatif serta lokasi-lokasi terjadinya kampanye.

Kata kunci : Bawaslu, JSON, kampanye, REST APIs

**ABSTRACT**

INDRIYANI. Development of Reporting Module on The Campaign Activity Applications for Society and Government Using REST APIs. Supervised by DEAN APRIANA RAMADHAN

Supervision of the campaign activity conducted by the Electoral Supervisory Agency (Bawaslu). Bawaslu has disadvantage on the implementing in the supervision that is carrying capacity obstacles which is limiting the number of Election Supervisors (PPL). Considering that obstacles required media for reporting that can be done by the society. This research developed a reporting module on the campaign activity applications for the society and the government using REpresentational State Transfer (REST) Application Programming Interface (API) with NoSQL database structure. The method used in this research is Extreme Programming. REST APIs is an architectural method of communication that was often applied in the development of web-based services. Inputting data at REST APIs using Node.js and the results of the data is the form of JSON text format. This research has successfully developed a web application that can help people to know about informaton from legislative candidates and the occurred locations of a campaign.

Keywords: Bawaslu, campaign, JSON, REST APIs

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

**PENGEMBANGAN MODUL PELAPORAN APLIKASI AKTIVITAS**

**KAMPANYE UNTUK MASYARAKAT DAN PEMERINTAH MENGGUNAKAN REST APIS**

**INDRIYANI**

**ILMU KOMPUTER**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**INSTITUT PERTANIAN BOGOR**

**BOGOR**

**2016**

Penguji: 1 Dr Yani Nurhadriyani, SSi MT

2 Irman Hermadi, SKom MS PhD

Judul Skripsi : Pengembangan Modul Pelaporan Aplikasi Aktivitas Kampanye untuk Masyarakat dan Pemerintah Menggunakan REST APIs

Nama : Indriyani

NIM : G64120028

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 skripsi dengan judul Pengembangan Modul Pelaporan Aplikasi Aktivitas Kampanye untuk Masyarakat dan Pemerintah Menggunakan REST APIsdapat diselesaikan. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Ilmu Komputer pada Fakultas Matematika dan Ilmu Pengetahuan Alam. Dalam penelitian ini banyak pihak yang telah membantu dan mendukung penulis dalam hal apapun. Oleh karena itu, penulis ingin menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Orang tua dan kakak penulis yang selalu memberikan doa, dukungan, dan motivasi kepada penulis.
2. Bapak Dean Apriana Ramadhan, SKomp MKom selaku dosen pembimbing yang selalu memberikan dukungan dan arahan serta selalu sabar dalam membimbing penulis.
3. Ibu Dr Yani Nurhadriyani, SSi MT dan Bapak Irman Hermadi, SKom MS PhD selaku dosen penguji.
4. Teman-teman satu bimbingan, yaitu Reza Pahlevi, Dieva Setyatama, Dian Setiana, Kadek Grihadevi, Moh Reza, dan Shellafuri untuk segala bentuk dukungannya.
5. Teman-teman kontrakan DM, yaitu Siti Rohaeni, Indah R Djukarsa, Arie Nurfita, Sri Wahyuni, Septi Dewi Cahaya, Aida Juniarti, dan Siti Khodijah yang setia menemani penulis selama tinggal di Bogor.
6. Ardhi Ma’arik, Eylien Desy Novita, Hilmi Salim, Juananda Dwi Satria, Kyulene Bunlya Fallah, Orita Mega Delani, dan Safira Widya Putri yang selalu menghibur dan menemani penulis.
7. Dan semua pihak lainnya yang terlibat dalam penulisan skripsi ini yang tidak dapat disebutkan namanya satu-persatu.

Penulis berharap semoga skripsi ini dapat bermanfaat bagi semua pihak yang membutuhkan.

Bogor, Agustus 2016

*Indriyani*

**DAFTAR ISI**

DAFTAR TABEL vi

DAFTAR GAMBAR vi

DAFTAR LAMPIRAN vii

[PENDAHULUAN 1](#_Toc459307639)

[Latar Belakang 1](#_Toc459307640)

[Perumusan Masalah 2](#_Toc459307641)

[Tujuan Penelitian 2](#_Toc459307642)

[Manfaat Penelitian 2](#_Toc459307643)

[Ruang Lingkup Penelitian 3](#_Toc459307644)

[TINJAUAN PUSTAKA 3](#_Toc459307645)

[*Roadmap* Penelitian 3](#_Toc459307646)

[Undang-Undang RI No. 1 Tahun 2015 4](#_Toc459307647)

[METODE PENELITIAN 4](#_Toc459307648)

[Data Penelitian 4](#_Toc459307649)

[Perancangan Arsitektur Perangkat Lunak 4](#_Toc459307650)

[Tahapan Penelitian 5](#_Toc459307651)

[Peralatan Penelitian 6](#_Toc459307652)

[HASIL DAN PEMBAHASAN 6](#_Toc459307653)

[Deskripsi Data 6](#_Toc459307654)

[Arsitektur Perangkat Lunak 7](#_Toc459307655)

[Iterasi Pertama 7](#_Toc459307656)

[Iterasi Kedua 14](#_Toc459307657)

[SIMPULAN DAN SARAN 21](#_Toc459307658)

[Simpulan 21](#_Toc459307659)

[Saran 22](#_Toc459307660)

[DAFTAR PUSTAKA 22](#_Toc459307661)

LAMPIRAN 24

RIWAYAT HIDUP 38

**DAFTAR TABEL**

1. [Data informasi lokasi kampanye 6](#_Toc459197322)
2. [*User story* masyarakat 8](#_Toc459197323)
3. [*User story* Bawaslu 9](#_Toc459197324)
4. [*Fully developed use case description* melaporkan pelanggaran 9](#_Toc459197325)
5. [Pengujian iterasi pertama REST API dengan *software* Postman 14](#_Toc459197326)
6. [Pengujian iterasi kedua REST API dengan *software* Postman 21](#_Toc459197327)

**DAFTAR GAMBAR**

1. [*Roadmap* bagian SEInS 3](#_Toc459307613)
2. [Alur pengembangan model *Extreme Programming* (Pressman 2010) 5](file:///G:\SKomp\Proposal\revisi\G64120028_Skripsi_01-08-16.docx#_Toc459307614)
3. [Arsitektur perangkat lunak 7](#_Toc459307615)
4. [*Use case* diagram iterasi pertama dari modul masyarakat 7](#_Toc459307616)
5. [*Use case* diagram iterasi pertama dari modul Bawaslu 8](#_Toc459307617)
6. [Pemodelan halaman melihat data laporan pelanggaran 10](#_Toc459307618)
7. [Kode koneksi REST dengan basis data 11](#_Toc459307619)
8. [Potongan kode REST API fungsi melaporkan pelanggaran dengan struktur database MySQL 11](#_Toc459307620)
9. [Format teks JSON ketika POST gagal atau MySQL tidak terkoneksi 12](#_Toc459307621)
10. [Format teks JSON ketika POST berhasil 12](#_Toc459307622)
11. [Potongan kode REST API melihat data laporan pelanggaran dengan struktur database MySQL 12](#_Toc459307623)
12. [Format teks JSON ketika GET berhasil 13](#_Toc459307624)
13. [Potongan kode REST API memberikan tindakan terhadap laporan pelanggran dengan struktur database MySQL 13](#_Toc459307625)
14. [Format teks JSON ketika PUT berhasil 13](file:///G:\SKomp\Proposal\revisi\G64120028_Skripsi_01-08-16.docx#_Toc459307626)
15. [*Use case* diagram iterasi kedua dari modul masyarakat 14](file:///G:\SKomp\Proposal\revisi\G64120028_Skripsi_01-08-16.docx#_Toc459307627)
16. [*Use case* diagram iterasi kedua dari modul Bawaslu 15](#_Toc459307628)
17. [*Swimlane* diagram memberikan tindakan terhadap laporan pelanggaran 15](#_Toc459307629)
18. [Pemodelan halaman Home pada *web* *browser* 16](#_Toc459307630)
19. [Potongan kode REST API fungsi *login* dengan struktur basis data NoSQL 17](#_Toc459307631)
20. [Potongan kode REST API fungsi melaporkan pelanggaran dengan struktur database NoSQL 17](#_Toc459307632)
21. [Format teks JSON untuk kode *error* 400 18](file:///G:\SKomp\Proposal\revisi\G64120028_Skripsi_01-08-16.docx#_Toc459307633)
22. [Format teks JSON untuk kode *error* 201 18](file:///G:\SKomp\Proposal\revisi\G64120028_Skripsi_01-08-16.docx#_Toc459307634)
23. [Potongan kode REST API fungsi melihat data laporan pelanggaran dengan struktur database NoSQL 18](#_Toc459307635)
24. [Potongan kode REST API memberikan tindakan terhadap laporan pelanggran dengan struktur database NoSQL 19](#_Toc459307636)
25. [Potongan kode html implementasi *task* melihat laporan 19](file:///G:\SKomp\Proposal\revisi\G64120028_Skripsi_01-08-16.docx#_Toc459307637)
26. [Potongan kode fungsi implementasi *task* melihat laporan 20](file:///G:\SKomp\Proposal\revisi\G64120028_Skripsi_01-08-16.docx#_Toc459307638)

**DAFTAR LAMPIRAN**

1. [Data laporan pelanggaran 24](#_Toc459197435)
2. [Data calon legislatif 25](#_Toc459197436)
3. [*Fully developed use case description* iterasi pertama melihat data laporan pelanggaran 26](#_Toc459197437)
4. [*Fully developed use case description* iterasi pertama melakukan *request* lokasi kampanye 27](#_Toc459197438)
5. [*Fully developed use case description* iterasi pertama memberikan tindakan terhadap laporan pelanggaran 28](#_Toc459197439)
6. [Rancangan basis datadengan struktur MySQL 29](#_Toc459197440)
7. [*Fully developed use case description* iterasi kedua melihat data laporan pelanggaran 30](#_Toc459197441)
8. [*Fully developed use case description* iterasi kedua melaporkan pelanggaran 31](#_Toc459197442)
9. [*Fully developed use case description* iterasi kedua melakukan *request* lokasi kampanye 32](#_Toc459197443)
10. [*Fully developed use case description* iterasi kedua memberikan tindakan terhadap laporan pelanggaran 33](#_Toc459197444)
11. [*Swimlane diagram task* melihat data laporan pelanggaran dan melaporkan pelanggaran 34](#_Toc459197445)
12. [*Swimlane diagram task* melakukan *request* lokasi kampanye 35](#_Toc459197446)
13. [Rancangan basis datadengan struktur NoSQL 36](#_Toc459197447)
14. [Implementasi pada *web* *browser* 37](#_Toc459197448)

# PENDAHULUAN

## Latar Belakang

Penerapan *Information* & *Communication Technology* (ICT) di pemerintahan merupakan upaya untuk mengembangkan penyelenggaraan pemerintahan yang berbasis elektronik dalam rangka meningkatkan transparansi dan kualitas pelayanan publik secara efektif dan efisien (Kominfo 2013). Salah satu contohpenerapan ICT dalam pemerintahan adalah *e-campaign*. *E-campaign* merupakan salah satu strategi dalam kampanye karena berkolerasi positif terhadap meningkatnya jumlah suara yang diperoleh (Ramadhan 2015). *E-campaign* membuat proses kampanye menjadi lebih interaktif yang memudahkan terjadinya interaksi antara calon legislatif dengan masyarakat melalui Internet. Menurut hasil riset yang dilakukan oleh Asosiasi Penyelenggara Jasa Internet Indonesia (APJII), pertumbuhan penggunaan Internet di Indonesia terus meningkat. Pada tahun 2014, survei menunjukkan pengguna Internet di Indonesia mencapai 82 juta jiwa atau 34.9% dari seluruh penduduk Indonesia. 40 juta jiwa atau 49% pengguna Internet di Indonesia berusia 18–25 tahun, yaitu masyarakat yang telah memiliki hak suara dalam Pemilu.

Kelembagaan Pengawasan Pemilu atau Panitia Pengawas Pelaksanaan Pemilihan Umum (Panwaslak Pemilu) pertama kali muncul pada saat pelaksanaan Pemilu tahun 1982. Pembentukkan Panwaslak dilatari oleh protes atas banyaknya pelanggaran dan manipulasi perhitungan suara yang dilakukan oleh para petugas Pemilu pada tahun 1971. Pada tahun 2007 melalui Undang-Undang Nomor 22 Tahun 2007 tentang Penyelenggaraan Pemilu, kelembagaan pengawas Pemilu dikuatkan dengan dibentuknya sebuah lembaga tetap yang dinamakan Badan Pengawas Pemilu (Bawaslu). Sesuai dengan Undang-Undang RI No. 1 Tahun 2015 tentang Penetapan Peraturan Pemerintah Pengganti Undang-Undang Nomor 1 Tahun 2014 tentang Pemilihan Gubernur, Bupati, dan Walikota Menjadi Undang-Undang, saat ini pengawasan pelaksanaan Pemilu masih dilaksanakan oleh Bawaslu.

Salah satu kendala yang dialami Bawaslu dalam pengawasan Pemilu adalah kendala daya dukung pengawasan berupa keterbatasan jumlah Pengawas Pemilu Lapangan (Bawaslu 2014). Jumlah laporan pelanggaran oleh masyarakat dan peserta pada Pemilu Kepala Daerah dan Wakil Kepala Daerah (Pemilukada) tahun 2013 adalah 1.001 laporan sedangkan pada Pemilu Legislatif tahun 2014 adalah 2.722 laporan. Meningkatnya jumlah laporan dari masyarakat menunjukkan bahwa masyarakat dapat berperan dalam pengawasan Pemilu.

Menurut Peraturan Badan Pengawasan Pemilihan Umum Nomor 14 Tahun 2012 tentang Tata Cara Pelaporan dan Penanganan Pelanggaran Pemilihan Umum Anggota Dewan Perwakilan Rakyat, Dewan Perwakilan Daerah, dan Dewan Perwakilan Rakyat Daerah, tata cara melaporkan pelanggaran dalam pengawasan Pemilu yaitu; (1) Pelapor wajib mengisi dan menandatangani formulir penerimaan Laporan Dugaan Pelanggaran Pemilu Model B.1-DD yang memuat nama dan alamat pelapor, waktu dan tempat peristiwa terjadi, nama dan alamat terlapor, nama dan alamat saksi-saksi, uraian kejadian, dan tanda tangan pelapor; (2) Menyertakan fotokopi Kartu Tanda Penduduk (KTP) atau kartu identitas lain. Bentuk laporan dugaan pelanggaran tersebut dapat dilakukan secara langsung ke kantor bawaslu atau secara tidak langsung dalam bentuk pesan singkat melalui telepon genggam, faksimili, surat elektronik, atau laporan di situs *web*/*website*.

Saat ini sudah banyak aplikasi yang dibuat untuk Pemilu, aplikasi-aplikasi tersebut dapat ditemukan di pemiluapps.org/aplikasi. Salah satu aplikasi yang menjadi perhatian adalah Pantau Pemilu. Dalam aplikasi Pantau Pemilu terdapat beberapa menu yaitu berita, peraturan, data kandidat, *polling*, pantau seputar Pemilu, grafik statistik hasil pantauan dari pengguna, data rekapitulasi KPU, dan hasil pemilihan di TPS.

Penelitian ini mengembangkan aplikasi *web* untuk pelaporan pelanggaran kampanye yang dilengkapi dengan penanda lokasi terjadinya kampanye serta masyarakat dapat meminta lokasi untuk dilakukan kampanye kepada calon legislatif. Perbedaan aplikasi Pantau Pemilu dengan aplikasi pada penelitian ini adalah pada menu kampanye. Dalam penelitian ini terdapat menu kampanye yang berisi informasi lokasi diadakannya kampanye serta masyarakat dapat melakukan *request* lokasi kampanye. Fokus pada penelitian ini adalah membangun REST APIs dari beberapa fungsi sehingga dapat digunakan pada *client* berbasis *website* dan *mobile*. Kelebihan dari penggunaan REST APIs adalah *client* tidak berhubungan langsung dengan basis data. REST APIs tersebut menjembatani komunikasi antara *client* dan *server*.

## Perumusan Masalah

Rumusan permasalahan pada penelitian ini yaitu bagaimana mengembangkan aplikasi modul pelaporan kampanye yang mampu memudahkan:

1. Masyarakat dalam melaporkan pelanggaran kampanye.
2. Masyarakat dalam mengetahui lokasi kampanye serta melakukan *request* lokasi diadakannya kampanye di daerah tertentu kepada calon legislatif.
3. Pemerintah dalam memberikan tanggapan terhadap laporan dari masyarakat.

## Tujuan Penelitian

Tujuan dari penelitian ini adalah mengembangkan sebuah aplikasi *web* aktivitaskampanye berbasis REST APIs yang memfasilitasi:

1. Masyarakat untuk melaporkan pelanggaran Pemilu yang bersifat anonim.
2. Masyarakat dalam mengetahui lokasi kampanye serta melakukan *request* lokasi diadakannya kampanye di daerah tertentu kepada calon legislatif.
3. Pemerintah dalam mendapatkan dan menanggapi laporan pelanggaran kampanye.

## Manfaat Penelitian

Manfaat yang diperoleh dari hasil penelitian ini adalah:

1. Dari sisi masyarakat
   1. Dapat melaporkan pelanggaran Pemilu secara cepat
   2. Memperoleh informasi seputar calon legislatif
   3. Memperoleh informasi lokasi kampanye serta dapat melakukan *request* lokasi diadakannya kampanye
2. Dari sisi pemerintah
   1. Lebih mudah dalam menanggapi laporan pelanggaran Pemilu
   2. Dapat mengarsipkan data secara *online*

## Ruang Lingkup Penelitian

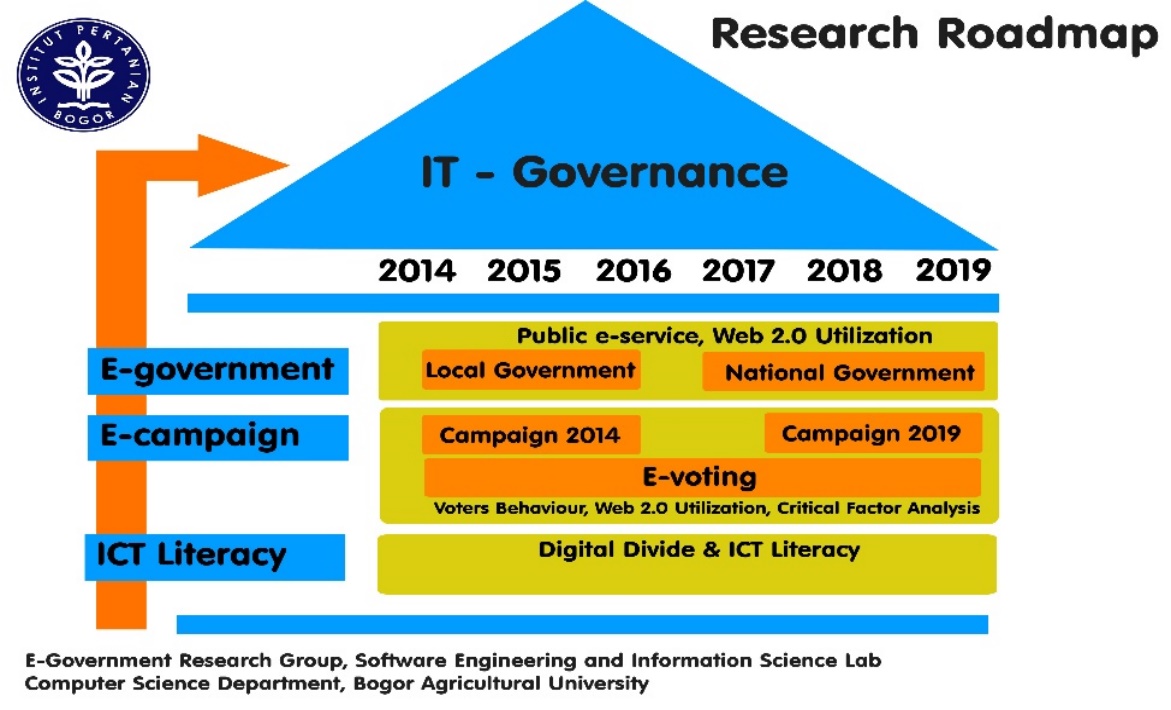
Ruang lingkup penelitian ini ialah:

1. Pengembangan modul pelaporan hanya dilakukan pada sisi pengguna masyarakat dan pemerintah berbasis *web* *browser*.
2. Tidak melakukan proses validasi terhadap data yang berasal dari masukan masyarakat.
3. REST APIs penelitian ini menggunakan JSON Web Token yang didapatkan pengguna saat melakukan login.
4. Implementasi pada *web* *browser* hanya untuk mengkonsumsi data dari REST APIs tanpa ada fasilitas untuk *sorting* dan *searching*.

# TINJAUAN PUSTAKA

## *Roadmap* Penelitian

*Roadmap* adalah sebuah rencana kerja yang mengintegrasikan seluruh rencana pelaksanaan program dalam kurun waktu tertentu. Laboratorium *Software Engineering and Information Science* (SEInS) *­e-government research group* memiliki *roadmap* untuk mencapai IT-*Governance* dengan tiga fokus rencana kerja yaitu ICT *literacy*, *e-campaign*, dan *e-government* dalam kurun waktu 2014−2019 yang dapat dilihat pada Gambar 1. Penelitian ini merupakan pelaksaan dari *roadmap* laboratorium SEInS pada rencana kerja *e-campaign* pada *web* 2.0 *utilization*. Pemantaan *web* 2.0 pada penelitian ini terdapat pada modul pelaporan pelanggaran dan *request* kampanye yang memungkinkan terjadinya interaksi antara pengguna dan sistem.



Gambar 1 *Roadmap* bagian SEInS

## Undang-Undang RI No. 1 Tahun 2015

Sesuai dengan Undang-Undang RI No. 1 Tahun 2015 Tentang Penetapan Peraturan Pemerintah Pengganti Undang-Undang Nomor 1 Tahun 2014 Tentang Pemilihan Gubernur, Bupati, dan Walikota Menjadi Undang-Undang pasal 69, dalam kampanye dilarang: (a) mempersoalkan dasar negara Pancasila dan Pembukaan Undang-Undang Dasar Negara Republik Indonesia Tahun 1945; (b) menghina seseorang, agama, suku, ras, golongan, Calon Gubernur, Calon Wakil Gubernur, Calon Bupati, Calon Wakil Bupati, Calon Walikota, Calon Wakil Walikota, dan/atau Partai Politik; (c) melakukan Kampanye berupa menghasut, memfitnah, mengadu domba Partai Politik, perseorangan, dan/atau kelompok masyarakat; (d) menggunakan kekerasan, ancaman kekerasan atau menganjurkan penggunaan kekerasan kepada perseorangan, kelompok masyarakat dan/atau Partai Politik; (e) mengganggu keamanan, ketenteraman, dan ketertiban umum; (f) mengancam dan menganjurkan penggunaan kekerasan untuk mengambil alih kekuasaan dari pemerintahan yang sah; (g) merusak dan/atau menghilangkan alat peraga Kampanye; (h) menggunakan fasilitas dan anggaran Pemerintah dan Pemerintah Daerah; (i) menggunakan tempat ibadah dan tempat pendidikan; (j) melakukan pawai yang dilakukan dengan berjalan kaki dan/atau dengan kendaraan di jalan raya; dan/atau (k) melakukan kegiatan Kampanye di luar jadwal yang telah ditetapkan oleh KPU Provinsi dan KPU Kabupaten/Kota. Menurut pasal 134 laporan pelanggaran Pemilihan disampaikan secara tertulis yang memuat paling sedikit: (a) nama dan alamat pelapor; (b) pihak terlapor; (c) waktu dan tempat kejadian perkara; dan (d) uraian kejadian

# METODE PENELITIAN

## Data Penelitian

Data yang digunakan dalam penelitian ini terdiri dari data yang berhubungan dengan Pemilu Legislatif seperti data pelanggaran beserta tanggapan pemerintah, data caleg, sebaran lokasi kampanye, dan *request* lokasi penyelenggaraan kampanye. Data pelanggaran dan *request* lokasi penyelenggaraan kampanye diperoleh dari masukan masyarakat. Data caleg diperoleh dari masukan *administrator*.Sebaran lokasi kampanye diperoleh dari masukan calon legislatif daerah. Data yang dimasukkan oleh *user* beracuan pada data Pemilu API *endpoint* Pemilu tahun 2014, sedangkan berita diperoleh dari situs-situs *online* berita seperti Liputan6, Okezone, Tempo, dan Antara News.

## Perancangan Arsitektur Perangkat Lunak

Perancangan arsitektur perangkat lunak memiliki tiga bagian yaitu data, *client side*, dan *server side*. Pada bagian data terdiri dari basis data dengan struktur MySql atau NoSQL, bagian *Client* terdapat aplikasi berbasis *website* dan *mobile android*, dan pada *server side* terdapat *Google Maps* API dan REST API dengan Node.js, dan pada bagian c*lient* terdapat aplikasi berbasis *website* dengan PHP *Frameworks* Code Igniter 3.0.3 dan *mobile android*.

## Tahapan Penelitian

Metode yang digunakan dalam penelitian ini adalah metode *Extreme Programming* yang merupakan salah satu model dari *Agile Software Development*. *Extreme Programming* adalah model pengembangan sistem perangkat lunak berbasis *Software Development Life Cycle* (SDLC) yang menyederhanakan berbagai tahapan dalam proses pengembangan sehingga pada Model XP proses menjadi lebih adaptif dan fleksibel. Menurut Mohammadi S *et al*. (2009) *Extreme**Programming* merupakan metode yang paling terkenal karena dalam praktiknya lebih menekankan pada pelanggan sehingga perubahan dapat dengan cepat ditanggapi. Proses pengembangan sistem pada model XP menurut Pressman (2010) terbagi menjadi empat tahapan utama seperti pada Gambar 2.

Tahapan yang dilakukan dalam penelitian ini adalah:

1. Perencanaan (XP *Planning*)

Tahap ini berfokus untuk mendapatkan gambaran fitur dan fungsi dari perangkat lunak yang akan dibangun. Pada tahap ini akan dibuat *use case* diagram, *user story*, *fully developed use case description* dan *swimlane* diagram dari modul masyarakat dan pemerintah.

1. Desain (XP *Design*)

Tahap ini bertujuan untuk mengatur pola logika dari sistem. Desain pada model proses *extreme programming* menjadi panduan dalam membangun perangkat lunak yang didasari pada hasil tahap *planning*.

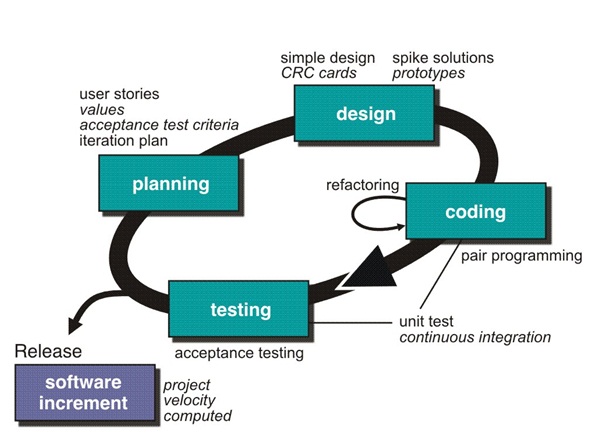
1. Pengkodean (XP *Coding*)

Setelah menyelesaikan perencanaan dan desain untuk aplikasi secara keseluruhan, XP lebih memprioritaskan terlebih dahulu untuk membuat modul unit tes yang bertujuan untuk melakukan uji coba setiap modul. Setelah berbagai unit tes selesai dibangun, kemudian dilakukan proses *coding* yang dilakukan permodul. Selanjutnya, modul aplikasi yang sudah selesai dibangun akan digabungkan dengan aplikasi utama.

1. Pengujian (XP *Testing*)

Walaupun pengujian telah dilakukan pada tahap *coding*, namun pada XP *Testing* ini akan memeriksa dan memperbaiki semua masalah-masalah yang terjadi.Setelah semua modul telah dikumpulkan dalam sebuah sistem yang sempurna, barulah pengujian penerimaan (*acceptance test*) dilakukan. Pada tahapan pengujian ini aplikasi langsung diuji coba oleh tim secara internal dan mendapat tanggapan langsung mengenai penerapan pada tahap perencanaan yang telah dilakukan sebelumnya.

Gambar 2  Alur pengembangan model *Extreme Programming* (Pressman 2010)



## Peralatan Penelitian

Penelitian ini dilakukan dengan menggunakan perangkat keras dan perangkat lunak sebagai berikut:

Perangkat keras berupa komputer personal dengan spesifikasi sebagai berikut:

1. Processor Intel® i3-4005U
2. RAM 4 GB
3. 500 GB HDD

Perangkat lunak:

1. Sistem operasi Windows 8.1
2. Bahasa pemrograman HTML, PHP, ajax, javascript, dan nodejs
3. Sublime Text sebagai *text editor*
4. Apache2 sebagai *web server*
5. DBMS MongoDB
6. Balsamiq sebagai *sofware* untuk membuat *mock-up*
7. Postman sebagai *sofware* untuk menguji REST APIs

# HASIL DAN PEMBAHASAN

## Deskripsi Data

Data contoh pada penelitian terdiri dari data laporan pelanggaran Pemilu tahun 2014, calon legislatif 2014 dan informasi lokasi kampanye. Data laporan pelanggaran Pemilu tahun 2014 dan calon legislatif 2014 diperoleh dari Pemilu API *endpoint* yang di unduh pada 12 April 2016 pada situs https://github.com/ pemiluAPI /pemilu-data. Data laporan pelanggaran dapat dilihat pada Lampiran 1. Dalam data tersebut terdapat atribut judul\_laporan, tanggal\_kejadian, alamat, keterangan, kategori, dan status. Pada penelitian ini atribut yang digunakan hanya judul\_laporan, tanggal\_kejadian, alamat, keterangan, kategori, dan status. Atribut keterangan digunakan untuk mendeskripsikan laporan sedangkan atribut kategori dan status untuk tanggapan pemerintah.

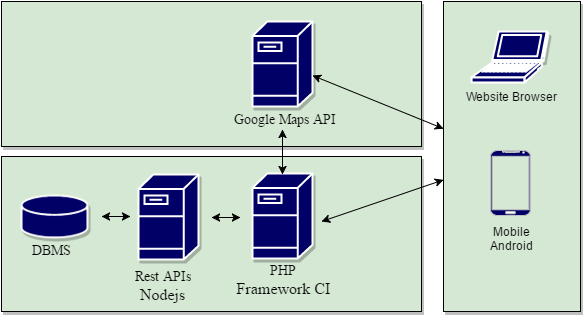
Data calon legislatif terdiri dari 12 atribut yaitu id, nama, alamat, lembaga, nama partai, no urut, jenis kelamin, agama, tanggal lahir, tempat lahir, dapil, dan pendidikan terakhir. Data calon legislatif tersebut dapat dilihat pada lampiran 2. Pada penelitian ini hanya diambil 7 calon legislatif dari 29566 calon legislatif yang ada pada Pemilu API. Data informasi lokasi kampanye memiliki 5 atribut yaitu nama\_caleg, lokasi, *latitude*, *longitude*, dan topik\_kampanye yang dapat dilihat pada Tabel 1.

Tabel 1  Data informasi lokasi kampanye

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nama\_caleg | Lokasi | *Latitude* | *Longitude* | Topik\_kampanye |
| R. Syaripudin | Cianjur, Jalan Astana Anyar, Nyengseret, Bandung City, West Java, Indonesia | -6.93085 | 107.6017 | Penyampaian visi dan misi |

## Arsitektur Perangkat Lunak

Analisis arsitektur perangkat lunak pada penelitian ini dapat dilihat pada Gambar 3. Penggunaan REST API dilakukan untuk mempermudah aliran data dari *client* dan *server*.REST API juga dapat digunakan oleh bahasa pemograman apapun sehingga dalam pengembangannya dapat digunakan oleh aplikasi berbasis *website* dan *mobile*. Penggunaan Google Maps API untuk memberikan penanda lokasi *longitude* dan *latitude* lokasi *request* kampanye, lokasi terjadinya kampanye dan lokasi terjadinya pelanggaran. Dalam implementasi nanti *website browser* sebagai *client* akan menggambil data dari DBMS melalui REST API dengan bahasa pemograman PHP pada *framework* Code Igniter.

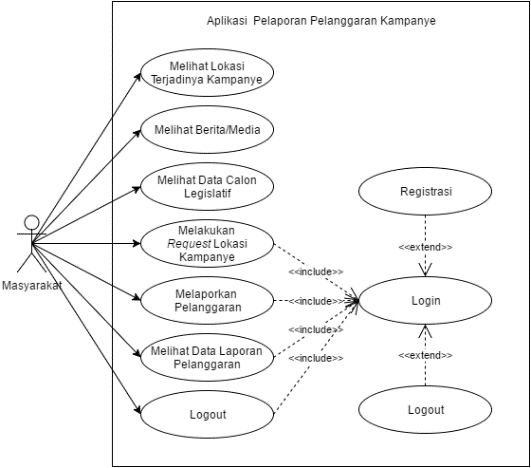


Gambar 3 Arsitektur perangkat lunak

## Iterasi Pertama

**Tahap Perencanaan (XP *Planning*)**

Pada tahap perencanaan iterasi pertama ini dilakukan pembuatan *use case* diagram dari modul masyarakat dan pemerintah yang bertujuan untuk mempermudah proses pembuatan desain pada tahap selanjutnya. *Use case* diagram dari modul masyarakat dapat dilihat pada Gambar 4 dan *use case* diagram dari modul pemerintah yang diwakili oleh Bawaslu dapat dilihat pada Gambar 5.



Gambar 4 *Use case* diagram iterasi pertama dari modul masyarakat



Gambar 5 *Use case* diagram iterasi pertama dari modul Bawaslu

Setelah membuat *use case* diagrammaka di buat *user story* dari *actor* masyarakat dan Bawaslu yang dapat dilihat pada Tabel 2 dan 3. Dalam *user story* masyarakat hanya terdiri dari tiga *task* utama dan Bawaslu terdiri dari satu *task* utama. Setelah *user story* selesai selanjutnya adalah membuat *fully developed use case description* dari semua *task* utama*. Fully developed use case description* dari *task* melaporkan pelanggaran dapat dilihat pada Tabel 4. Dalam *fully developed use case description* tersebut menjelaskan perihal *use case name*, *scenario task*, *triggering event*, *brief condition*, *actors*, *related use case*, *stakeholders*, *precondition*, *postconditions*, *flow of activities*, dan *exception condition* dari *task* melaporkan pelanggaran. *Fully developed use case description* dari *task* yang lain dapat dilihat pada Lampiran 3 – 5.

Tabel 2  *User story* masyarakat

|  |  |  |  |
| --- | --- | --- | --- |
| *Actor* | *Task* | *User story* | *Priority* |
| Masyarakat | Melaporkan pelanggaran | Masyarakat dapat melaporkan pelangaran-pelanggaran yang dilihat saat kampanye berlangsung secara cepat | 1 |
|  | Melihat data laporan pelanggaran | Masyarakat dapat melihat data laporan pelanggaran yang telah dilaporkan oleh masyarakat lainnya serta ingin mengetahui tindakan apa yang dilakukan oleh Bawaslu terkait dengan laporan tersebut | 2 |
|  | Melakukan *request* lokasi kampanye | Masyarakat dapat Calon Legislatif melakukan kampanye di daerah yang masyarakat inginkan | 3 |

Tabel 3  *User story* Bawaslu

|  |  |  |  |
| --- | --- | --- | --- |
| *Actor* | *Task* | *User story* | *Priority* |
| Bawaslu | Memberikan tindakan terhadap laporan pelanggaran | Bawaslu ingin memberikan tindakan terhadap laporan pelanggaran yang telah dilaporkan oleh masyarakat | 1 |

Tabel 4  *Fully developed use case description* melaporkan pelanggaran

|  |  |
| --- | --- |
| *Use case Name:* | Melaporkan pelanggaran |
| *Scenario:* | Masyarakat ingin melaporkan pelanggaran |
| *Triggering Event:* | Masyarakat memilih menu Pantau lalu Input Laporan |
| *Brief Description:* | Melaporkan pelangaran-pelanggaran yang dilihat saat kampanye berlangsung |
| *Actors:* | Masyarakat |
| *Related Use Case:* | *Login* masyarakat |
| *Stakeholders:* | Masyarakat |
| *Preconditions:* | Form *input* pelanggaran sudah ditampilkan |
| *Postconditions:* | Masyarakat berhasil melaporkan laporan pelanggaran |
| *Flow of Activities:* | |  |  | | --- | --- | | *Actor* | *System* | | 1. Memilih menu Pantau 2. Memilih menu Input Laporan 3. Mengisi form memasukkan pelanggaran 4. Pilih ‘Simpan’ pada halaman sistem | 2.1 Menampilkan form Input Laporan  4.1 Data laporan pelanggaran telah masuk ke dalam API Pelanggaran | |
| *Exception Conditions:* | Tidak ada kondisi |

**Tahap Desain (XP *Design*)**

Pada tahap ini dilakukan pembuatan pemodelan halaman dari *task* yang terdapat pada *user story*. Gambar 6 merupakan pemodelan halaman dari fungsi melihat data laporan pelanggaran. Pemodelan halaman dibuat agar pembuatan *web* aplikasi tidak menyimpang dari tujuan awal. Pada persegi panjang berwarna oranye ( ) terdapat beberapa menu navigasi yaitu home, kandidat, pantau, kampanye, media, dan peraturan. Pada Gambar 6 menu navigasi yang aktif adalah menu pantau. *Content* dari menu pantau ditandai dengan persegi panjang berwarna biru ( ). Dalam *content* terdapat beberapa data laporan pelanggaran yang terdiri dari judul, tanggal, isi, tanggapan Bawaslu dan komentar publik.



Gambar 6 Pemodelan halaman melihat data laporan pelanggaran

Pada tahap ini juga dilakukan perancangan basis data beracuan pada data Pemilu 2014 yang terdapat pada situs API Pemilu. Rancangan basis data pada iterasi pertama ini menggunakan struktur basis dataMySQL yang dapat dilihat pada Lampiran 6.

**Tahap Pengkodean (XP *Coding*)**

Dalam konsep REST ini aplikasi atau *client side* tidak terhubung langsung dengan basis data. Aplikasi memanggil data melalui REST, lalu REST lah yang berhubungan dengan basis data. Gambar 7 merupakan kode koneksi antara REST dengan basis data. Pada kode tersebut REST dikoneksikan dengan *host* adalah localhost, nama basis datanya adalah pemilukita, *base route* adalah apipemilukita, dan *port server* yang digunakan adalah 3003. Saat server REST diaktifkan akan muncul pesan “All right ! I am alive at Port 3003.”.

Pada tahap pengkodean telah dilakukan proses pembuatan REST API dari fungsi melaporkan pelanggaran, *request* kampanye, dan memberi tanggapan terhadap laporan dengan struktur basis data menggunakan MySQL. Fungsi REST API yang dibuat yaitu POST, GET, dan PUT. Gambar 8 merupakan potongan kode REST dari fungsi melaporkan pelanggaran. Pada potongan kode REST ini *method* yang dilakukan adalah POST. *Method* POST digunakan untuk memasukan *record* ke dalam tabel t\_laporan dengan *field* id\_masyarakat, id\_caleg, pesan, bukti, *parent*, latitude, longitude, dan tanggal yang dimasukkan oleh masyarakat.

13 REST**.prototype.**connectMysql **=** **function()** **{**

14 **var** **self** **=** **this;**

15 **var** pool **=** mysql**.**createPool**({**

16 connectionLimit **:** 100**,**

17 host **:** 'localhost'**,**

18 user **:** 'root'**,**

19 **password** **:** ''**,**

20 database **:** 'pemilukita'**,**

21 debug **:** **false**

22 **});**

23 pool**.**getConnection**(function(**err**,**connection**){**

24 **if(**err**)** **{**

25 **self.**stop**(**err**);**

26 **}** **else** **{**

27 **self.**configureExpress**(**connection**);**

28 **}**

29 **});**

30 **}**

31

32 REST**.prototype.**configureExpress **=** **function(**connection**)** **{**

33 **var** **self** **=** **this;**

34 app**.**use**(**bodyParser**.**urlencoded**({** extended**:** **true** **}));**

35 app**.**use**(**bodyParser**.**json**());**

36 **var** router **=** express**.**Router**();**

37 app**.**use**(**'/apipemilukita'**,** router**);**

38 **var** rest\_router **=** **new** rest**(**router**,**connection**,**md5**);**

39 **self.**startServer**();**

40 **}**

41

42 REST**.prototype.**startServer **=** **function()** **{**

43 app**.**listen**(**3003**,function(){**

44 console**.**log**(**"All right ! I am alive at Port 3003."**);**

45 **});**6 **}**

Gambar 7 Kode koneksi REST dengan basis data

385 router**.**post**(**"/laporan"**,function(**req**,**res**){**

386 **var** query **=** "INSERT INTO ??(??,??,??,??,??,??,??,??) VALUES (?,?,?,?,?,?,?,?)"**;**

387 **var** table **=** **[**"t\_laporan"**,**"id\_masyarakat"**,**"id\_caleg"**,**"pesan"**,**"bukti"**,**"parent"**,**"latitude"**,**"longitude"**,**"tanggal"**,**req**.**body**.**id\_masyarakat**,**req**.**body**.**id\_caleg**,**req**.**body**.**pesan**,**req**.**body**.**bukti**,**req**.**body**.parent,**req**.**body**.**latitude**,**req**.**body**.**longitude**,**req**.**body**.**tanggal**];**

388 query **=** mysql**.**format**(**query**,**table**);**

389 connection**.**query**(**query**,function(**err**,**rows**){**

390 **if(**err**){**

391 res**.**json**({**"Error" **:** **true,** "Message" **:** "Error executing MySQL query"**});**

392 **}else{**

393 res**.**json**({**"Error" **:** **false,** "Message" **:** "Laporan Added !"**});**

394 **}**

395 **});**

396 **});**

Gambar 8 Potongan kode REST API fungsi melaporkan pelanggaran dengan struktur database MySQL

REST dari fungsi melaporkan pelanggaran melaporkan pelanggaran dapat dijalankan dengan mengakses alamat pada url http://localhost:3003/apipemilukita/ laporan dengan *method* POST. Dalam proses melaporkan pelanggaran jika MySQL tidak terkoneksi maka terdapat *error* dan akan muncul keluaran berupa format teks JSON seperti pada Gambar 9. Jika MySQL telah terkoneksi maka *record* pelaporan akan tersimpan ke dalam tabel t\_laporan dan menghasilkan keluaran seperti pada Gambar 10.

1 {

2 "Error": true,

3 "Message": " Error executing MySQL query "

4 }

Gambar 9 Format teks JSON ketika POST gagal atau MySQL tidak terkoneksi

1 {

2 "Error": false,

3 "Message": " Laporan Added ! "

4 }

Gambar 10 Format teks JSON ketika POST berhasil

Kode REST dari *method* GET terdapat pada fungsi melihat data laporan pelanggaran seperti pada Gambar 11. Pada kode ini dilakukan pengambilan data yang ada pada tabel t\_laporan. Seperti pada proses POST, jika MySQL tidak terkoneksi maka akan muncul keluaran dengan *message* “Error executing MySQL query”. Jika MySQL telah terkoneksi akan muncul keluaran berupa *record* dari tabel t\_laporan dalam format teks JSON seperti pada Gambar 12.

398 router**.**get**(**"/laporan"**,function(**req**,**res**){**

399 **var** query **=** "SELECT \* FROM ??"**;**

400 **var** table **=** **[**"t\_laporan"**];**

401 query **=** mysql**.**format**(**query**,**table**);**

402 connection**.**query**(**query**,function(**err**,**rows**){**

403 **if(**err**){**

404 res**.**json**({**"Error" **:** **true,** "Message" **:** "Error executing MySQL query"**});**

405 **}else{**

406 res**.**json**({**"Error" **:** **false,** "Message" **:** "Success"**,** "Laporan" **:** rows**});**

407 **}**

408 **})**

409 **});**

Gambar 11 Potongan kode REST API melihat data laporan pelanggaran dengan struktur database MySQL

Dalam format teks JSON pada Gambar 12, REST API tanpa *error* dengan status berhasil dan memiliki *array* berupa laporan yang berisi satu objek dengan 10 *field* yang ada pada tabel t\_laporan. Objek pada format teks JSON berada di dalam kurung siku setelah nama *array*. Antar objek dalam *array* dipisahkan dengan kurung kurawal. Format teks JSON ini didapatkan dengan mengakses alamat url http://localhost:3003/ apipemilukita/laporan dengan *method* GET.

4 "Laporan": [

5 {

6 "id\_laporan": 7,

7 "id\_masyarakat": 10,

8 "id\_caleg": 3,

9 "title": "Kampanye PPP di Pamulang melibatkan anak-anak",

10 "pesan": "Kampanye Partai Persatuan Pembangunan (PPP) di Pamulang, Tangerang Selatan telah melibatkan anak-anak. Kampanye tersebut dilakukan pada hari Senin, 30 Maret 2014 pukul 09.00. Dari pukul 09.00- 10.00 WIB kerumunan massa PPP berkumpul di perumahan Griya Jakarta, Pamulang. Berdasarkan peraturan KPU, setiap kampanye terbuka partai politik dilarang melibatkan anak-anak. Disertakan beberapa foto dalam laporan ini.",

11 "bukti": "ff",

12 "parent": "Status laporan telah di verifikasi dan termasuk dalam kategori dugaan Pelanggaran Administratif",

13 "latitude": "4",

14 "longitude": "5",

15 "tanggal": "2014-03-30T17:00:00.000Z"

16 }

17 ]

18 }

Gambar 12 Format teks JSON ketika GET berhasil

Kode REST dari *method* PUT terdapat pada fungsi memberikan tindakan terhadap laporan pelanggaran seperti pada Gambar 13. Pada saat pengguna melakukan POST laporan, isi dari *field* *parent* adalah “Belum ada tanggapan” karena belum ada tanggapan dari Bawaslu. Jika Bawaslu melakukan tanggapan atau mengakses alamat url http://localhost:3003/ apipemilukita/laporan dengan *method* PUT maka *field parent* atau tanggapan dalam tabel t\_laporan akan berubah. Format keluaran dari REST API memberikan tindakan terhadap laporan adalah seperti pada Gambar 14.

424 router**.**put**(**"/laporan"**,function(**req**,**res**){**

425 **var** query **=** "UPDATE ?? SET ?? = ? WHERE ?? = ?"**;**

426 **var** table **=** **[**"t\_laporan"**,**"parent"**,**req**.**body**.parent,**"id\_laporan"**,**req**.**body**.**id\_laporan**];**

427 query **=** mysql**.**format**(**query**,**table**);**

428 connection**.**query**(**query**,function(**err**,**rows**){**

429 **if(**err**)** **{**

430 res**.**json**({**"Error" **:** **true,** "Message" **:** "Error executing MySQL query"**});**

431 **}** **else** **{**

432 res**.**json**({**"Error" **:** **false,** "Message" **:** "Updated tanggapan"**});**

433 **}**

434 **});**

435 **});**

Gambar 13 Potongan kode REST API memberikan tindakan terhadap laporan pelanggran dengan struktur database MySQL

1 {

2 "Error": true,

3 "Message": " Updated tanggapan "

4 }

Gambar 14 Format teks JSON ketika PUT berhasil

**Tahap Pengujian (XP *Testing*)**

Tahap pengujian iterasi pertama dilakukan pada aplikasi Postman. Pengujian ini bertujuan untuk memeriksa apakah REST API berfungsi atau tidak. Pengujian iterasi pertama ini dilakukan secara internal oleh tim dan hasil dari pengujian REST yang diperoleh pada tahap pengkodean berfungsi dengan baik. Hasil dari pengujian iterasi pertama ini dapat dilihat pada Tabel 5.

Tabel 5  Pengujian iterasi pertama REST API dengan *software* Postman

|  |  |  |
| --- | --- | --- |
| *Actor* | Fungsi REST | Status |
| Masyarakat | *POST* Laporan  *GET*  Laporan  *PUT* Laporan | Berhasil  Berhasil  Berhasil |

## Iterasi Kedua

**Tahap Perencanaan (XP *Planning*)**

Tahap perencanaan iterasi kedua dilakukan pemeriksaan dan perbaikan dari *use case* diagram dan *fully developed use case description*. Gambar 15 dan 16 merupakan bentuk dari perbaikan *use case* dimana untuk mengakses data pada REST masyarakat harus melakukan *login* terlebih dahulu untuk mendapatkan *token authentication*. Token yang digunakan pada penelitian ini adalah JSON Web Token (JWT). Perbaikan dari *fully developed use case* dapat dilihat pada Lampiran 7 − 10. Pada iterasi ini terjadi perubahan struktur basis data dari MySQL menjadi NoSQL yaitu MongoDB. Perubahan struktur basis data ini dilakukan karena MongoDB lebih efisien dan performa yang ditawarkan lebih cepat dibandingkan MySQL.

Gambar 15 *Use case* diagram iterasi kedua dari modul masyarakat

 Gambar 16 *Use case* diagram iterasi kedua dari modul Bawaslu

Pada tahap perencanaan ini juga di buat *swimlane* diagramyang berfungsi untuk mengetahui aktor yang terlibat dalam suatu *task*. Dari semua *task* utama hanya *task* memberikan tindakan terhadap laporan pelanggaran saja yang melibatkan dua aktor yaitu masyarakat dan Bawaslu yang dapat dilihat pada Gambar 17. *Swimlane* dari *task* utama yang lainnya dapat dilihat pada Lampiran 11 dan 12.



Gambar 17 *Swimlane* diagram memberikan tindakan terhadap laporan pelanggaran

**Tahap Desain (XP *Design*)**

Pada iterasi ini dilakukan pemodelan halaman dari fungsi utama langsung pada tampilan *web*. Gambar 18 merupakan pemodelan halaman Home pada *web* *browser*. Menu navigasi pada pemodelan ini ditandai dengan persegi panjang berwarna oranye ( ) dan *content* ditandai dengan persegi berwarna biru ( ). Pada iterasi ini juga dilakukan perangcangan basis data, untuk struktur basis data NoSQL yang dapat dilihat pada Lampiran 13. Pada struktur basis data tersebut terdapat 8 *schema collections* yaitu *user*, admin, masyarakat, caleg, pemerintah, laporan, *request*, dan kampanye. *Schema user* memiliki 5 *field*, *schema* admin memiliki 3 *field*, *schema* masyarakat memiliki 6 *field*, *schema* caleg memiliki 14 *field*, *schema* pemerintah memiliki 6 *field*, *schema* laporan memiliki 10 *field*, dan *schema* kampanye memiliki 6 *field*.



Gambar 18 Pemodelan halaman Home pada *web* *browser*

**Tahap Pengkodean (XP *Coding*)**

Tahap pengkodean iterasi kedua adalah pembuatan REST API menggunakan struktur basis data NoSQL yaitu MongoDB. Sama seperti pada iterasi pertama, pada tahap pengkodean iterasi pertama ini juga dibuat REST dengan tiga *method* yaitu POST, GET, dan PUT. Saat pemanggilan REST API diperlukan *authenticate* atau *token* yang diberikan pada saat login, sehingga REST API tidak dapat di akses oleh pengguna yang tidak berhasil melakukan *login*. Kode REST API untuk mendapatkan *authenticate* dapat dilihat pada Gambar 19.

Potongan kode REST dengan *method* POST dapat dilihat pada Gambar 20. URL API berada pada *path* /laporan sehingga URL dari REST API ini adalah http://localhost:4040/api/laporan. Pada sturktur REST dalam pengkodean iterasi kedua ini server menggunakan *host* adalah localhost, *base route* adalah api, dan port yang digunakan adalah 4040.

101 apiRoutes**.**post**(**'/authenticate'**,** **function(**req**,** res**)** **{**

102 User**.**findOne**({**

103 email**:** req**.**body**.**email

104 **},** **function(**err**,** user**)** **{**

105 **if** **(**err**)** **throw** err**;**

106

107 **if** **(!**user**)** **{**

108 res**.status(**401**).**json**({** success**:** **false,** message**:** 'Authentication failed. User not found.' **});**

109 **}** **else** **{**

110 // Check if password matches

111 user**.**comparePassword**(**req**.**body**.password,** **function(**err**,** isMatch**)** **{**

112 **if** **(**isMatch **&&** **!**err**)** **{**

113 // Create token if the password matched and no error was thrown

114 **const** token **=** jwt**.**sign**(**user**,** config**.**secret**,** **{**

115 expiresIn**:** 10080 // in seconds

116 **});**

117 res**.status(**200**).**json**({** success**:** **true,** token**:** 'JWT ' **+** token **,** data**:** user**});**

118 **}** **else** **{**

119 res**.status(**401**).**json**({** success**:** **false,** message**:** 'Authentication failed. Passwords did not match.' **});**

120 **}**

121 **});**

122 **}**

123 **});**

124 **});**

Gambar 19 Potongan kode REST API fungsi *login* dengan struktur basis data NoSQL

389 apiRoutes**.**post**(**'/laporan'**,** requireAuth**,** **function(**req**,** res**)** **{**

390 console**.**log**(**req**.**body**);**

391 **if(!**req**.**body**.**id\_user **||** **!**req**.**body**.**nama\_caleg **||** **!**req**.**body**.**judul\_laporan **||** **!**req**.**body**.**pesan **||** **!**req**.**body**.**lokasi **||** **!**req**.**body**.**lat **||** **!**req**.**body**.**lng **||** **!**req**.**body**.**tanggal**)** **{**

392 res**.status(**400**).**json**({** success**:** **false,** message**:** 'Please enter data.' **});**

393 **}** **else** **{**

394 **const** newLaporan **=** **new** Laporan**({**

395 id\_user**:** req**.**body**.**id\_user**,**

396 nama\_caleg**:** req**.**body**.**nama\_caleg**,**

397 judul\_laporan**:** req**.**body**.**judul\_laporan**,**

398 pesan**:** req**.**body**.**pesan**,**

399 bukti**:** 'http://localhost:4040/'+req.file.originalname

400 lokasi**:** req**.**body**.**lokasi**,**

401 lat**:** req**.**body**.**lat**,**

402 lng**:** req**.**body**.**lng**,**

403 tanggapan**:**"Belum ada tanggapan"**,**

404 tanggal**:** req**.**body**.**tanggal

405 **});**

406 // Attempt to save the user

407 newLaporan**.**save**(function(**err**)** **{**

408 **if** **(**err**)** **{**

409 **return** res**.status(**400**).**json**({** success**:** **false,** message**:** 'That laporan already exists.'**});**

410 **}**

411 res**.status(**201**).**json**({** success**:** **true,** message**:** 'Successfully created laporan.' **});**

412 **});**

413 **}**

414 **});**

Gambar 20 Potongan kode REST API fungsi melaporkan pelanggaran dengan struktur database NoSQL

Pada proses POST terdapat delapan *field* yang harus diisi yaitu id\_user, nama\_caleg, judul\_laporan, pesan, lokasi, lat, lng, dan tanggal. Jika diantara kedelapan *field* tersebut tidak diisi maka akan terjadi *error* 400 dengan menampilkan format teks JSON seperti pada Gambar 21. Jika kedelapan *field* telah terisi dan sesuai dengan ketentuan dalam model *schema* tabel maka masukkan akan disimpan dalam basis data mongodb://127.0.0.1:27017/pemilukita pada *collections* laporans dengan keluaran format JSON seperti pada Gambar 22 dengan status *error* 201 atau *created*.

Gambar 21 Format teks JSON untuk kode *error* 400

1 {

2 "success": false,

3 "message": "Successfully created laporan."

4 }

1 {

2 "success": false,

3 "message": "Please enter data."

4 }

Pengkodean REST API dengan *method* GET pada struktur basis dataMongoDB dapat dilihat pada Gambar 23. Pada potongan kode REST API GET ini dilakukan pencarian *record* pada *schema* atau tabel *collections* laporans. Jika tidak ditemukan *error* maka akan muncul keluaran berupa format teks JSON dengan *array* laporan.

Gambar 22 Format teks JSON untuk kode *error* 201

416 apiRoutes**.**get**(**'/laporan'**,** requireAuth**,** **function(**req**,** res**)** **{**

417 Laporan**.**find**({},** **function(**err**,** laporan**)** **{**

418 **if(**err**){**

419 res**.status(**400**).**send**(**err**);**

420 **}else{**

421 res**.status(**200**).**json**({**laporan**});**

422 **}**

423 **});**

424 **});**

Gambar 23 Potongan kode REST API fungsi melihat data laporan pelanggaran dengan struktur database NoSQL

Pengkodean REST API dengan *method* PUT pada struktur basis dataMongoDB dapat dilihat pada Gambar 24. Pada potongan kode REST API PUT ini dilakukan pencarian *record* pada *schema* atau tabel *collections* laporans dengan id\_laporan tertentu kemudian dilakukan perubahan pada *field* tanggapan. Jika tidak ditemukan *error* maka akan muncul keluaran berupa format teks JSON dengan *message* “Laporan telah di tanggapi!”.

453 apiRoutes**.**put**(**'/laporan'**,** requireAuth**,** **function(**req**,** res**)** **{**

454 Laporan**.**findOneAndUpdate**({**'\_id'**:** req**.**body**.**id\_laporan**},** **{**$set**:{**tanggapan**:**req**.**body**.**tanggapan**}},function(**err**,** laporan**)** **{**

455 **if** **(**err**)**

456 res**.**send**(**err**);**

457

458 res**.**json**({** message**:** 'Laporan telah di tanggapi!' **});**

459 **});**

460 **});**

Gambar 24 Potongan kode REST API memberikan tindakan terhadap laporan pelanggran dengan struktur database NoSQL

Pada iterasi ini juga dilakukan implementasi dari fungsi REST yang telah dibuat menjadi *web* dengan menggunakan *framework* Code Igniter. Bahasa pemograman yang digunakan adalah *Asynchronous JavaScript* *and XMLHTTP* (AJAX) agar aplikasi *web* menjadi lebih interaktif dan responsif serta memiliki kemampuan proses *request* ke *server* yang cepat. Gambar 25 merupakan kode html dan Gambar 26 merupakan potongan kode fungsi dari implementasi task melihat laporan.

Gambar 25 Potongan kode html implementasi *task* melihat laporan

300 <!--page pelanggaran-->

301 <div id=**"laporan"** style=**"display:none;"**>

302 <div class=**"row"**>

303<div class=**"box"**>

304<div class=**"col-lg-12"**>

305<h2 class=**"intro-text text-center"**>

306<div class=**"btn-row"**>

307<input type=**"hidden"** id=**"id\_profil\_l"** value=**"**<?php **echo** $this->session->userdata('id'); ?>**"**>

308<button type=**"button"** class=**"btn btn-info"** onclick=**"create\_laporan()"**><i class=**"glyphicon glyphicon-pencil"**></i> **Buat Laporan**</button></button>

309<button type=**"button"** class=**"btn btn-info"** onclick=**"laporansaya()"**><i class=**"glyphicon glyphicon-file"**></i> **Laporan Saya**</button></button>

310</div>

311</h2>

312<header class=**"panel-heading"**>

313<h1>**Laporan Pelanggaran**</h1>

314</header><br/>

315<section class=**"table1"**>

316<table border=**"0"**>

317<tbody id=**"body1"**>

318</tbody>

319</table>

320</section>

321</div>

322</div>

323</div>

324 </div>

Implementasi pada potongan kode html ini berfungsi untuk melakukan *parsing* dari format teks JSON menjadi tampilan *web* sehingga dapat *user* dapat mengerti data tersebut. Potongan kode fungsidigunaan untuk memanggil url dari REST API yang kemudian akan ditampilkan dalam bentuk *web* oleh kode html. Pada potongan kode fungsi Gambar 26 yang dilakukan adalah *method* GET pada url http://localhost:4040/api/laporan atau menampilkan data laporan pelanggaran. Tampilan *web* aplikasi dapat dilihat pada Lampiran 14.

760 ***function*** laporan() {

761 $.ajax({

762 type: 'GET',

763 url: "http://localhost:4040/api/laporan",

764 //mengirim token

765 beforeSend: function(xhr){

766 xhr.setRequestHeader('Authorization', localStorage.token);

767 },

768 success: ***function***(msg){

769 $("#welcome").hide();

770 $("#berita").hide();

771 $("#1").hide();

772 $("#create\_laporan").hide();

773 $("#detail\_laporan").hide();

774 $("#laporansaya").hide();

775 $("#my\_request").hide();

776 $("#kampanye").hide();

777 $("#create\_request").hide();

778 $("#caleg").hide();

779 $("#detail\_caleg").hide();

780 $("#edit\_profil").hide();

781 $("#my\_profil").hide();

782 $("table tbody#body1 tr").remove();

783 $("#laporan").show();

784 $(msg.laporan).each(***function***(key, value){

key++;

785 $("table #body1").prepend("<tr><input type='hidden' id='detail"+key+"' value="+value.\_id+"><th><h4>"+value.judul\_laporan+"</h4></th></tr><tr><th>"+value.tanggal+"</th></tr><tr><th>"+value.pesan+"</th></tr><tr><th><br/>Tanggapan: "+value.tanggapan+" </th></tr><tr><th><div class='btn-group'><button class='btn btn-info btn-xs' onclick='detail\_laporan("+key+")'>Detail Laporan</button></div></th></tr><tr><th><hr></th></tr>");

786 });

787 }

788 });

789 }

*Pair programming* pada iterasi ini dilakukan menggunakan Git. Dengan menggunakan Git, setiap orang dalam sebuah tim dapat melakukan perubahan pada *source code* tanpa harus takut terjadi bentrok ataupun kesulitan dalam menggabungkan hasil perubahan yang dilakukan.

Gambar 26 Potongan kode fungsi implementasi *task* melihat laporan

**Tahap Pengujian (XP *Testing*)**

Tahap pengujian iterasi kedua dilakukan pengujian REST API yang menggunakan arsitektur basis dataNoSQL dengan *software* Postman. Tabel 3 merupakan pengujian dari fungsi REST API *user* masyarakat dan Bawaslu dengan menggunakan *software* Postman yang dilakukan oleh internal tim. Setelah pengujian REST API berhasil tahap selanjutnya adalah pengujian terhadap implementasi REST API pada aplikasi *web*.

Tabel 6  Pengujian iterasi kedua REST API dengan *software* Postman

|  |  |  |
| --- | --- | --- |
| *Actor* | Fungsi REST | Status |
| Masyarakat | *POST* *Register*  *POST Authenticate* (*login*)  *POST Profile* Masyarakat  *GET All* Laporan  *GET* Laporan *by Id User*  *POST* Laporan  *GET All Request* kampanye  *GET Request* Kampanye *by Id*  *POST Request* Kampanye  *GET All* Kampanye  *GET* Kampanye *by User* (caleg)  *GET Profile* Caleg  *GET Logout* | Berhasil  Berhasil  Berhasil  Berhasil  Berhasil  Berhasil  Berhasil  Berhasil  Berhasil  Berhasil  Berhasil  Berhasil  Berhasil |
| Bawaslu | *POST Authenticate* (*login*)  *POST Profile* Bawaslu  *PUT* Laporan *by Id* (memberi tanggapan)  *GET All* Kampanye  *GET All* Laporan  *GET Profile* Caleg  *GET Logout* | Berhasil  Berhasil  Berhasil  Berhasil  Berhasil  Berhasil  Berhasil |

# SIMPULAN DAN SARAN

## Simpulan

Penelitian ini telah berhasil mengembangkan aplikasi *web* berbasis REST APIs yang dikembangkan dengan *framework* Code Igniter versi 3.0.3. Pembuatan REST APIs menggunakan bahasa pemograman Node.js dan struktur basis dataNoSQL yaitu MongoDB. Aplikasi *web* ini dapat membantu proses kampanye menjadi lebih interaktif antara masyarakat dan calon legislatif dengan adanya data lokasi kampanye. Aplikasi *web* ini juga dapat menjadi fasilitas masyarakat dalam melaporkan pelanggaran yang mereka temui dalam pelaksanaan kampanye Pemilu legislatif sehingga membantu Bawaslu dalam pengawasan dan mendapatkan data laporan pelanggaran secara *online*.

## Saran

Penelitian ini dapat dikembangkan dengan menambahkan fungsi *sharing* kegiatan kampanye kepada masyarakat agar dapat mengoptimalkan pelaksanaan kampanye Pemilu legislatif secara *online*. Aplikasi *web* ini juga dapat dikembangkan dengan menambahkan teknik pengambilan keputusan agar data laporan yang ditampilkan adalah data sebenarnya dan tidak memiliki unsur *black campaign*.

# DAFTAR PUSTAKA

[APJII] Asosiasi Penyelenggara Jasa Internet Indonesia. 2014. Profil Pengguna Internet Indonesia [Internet]. [diunduh 2015 November 1]. Tersedia pada: http://www.slideshare.net/internetsehat/profil-pengguna-internet-indonesia-2014-riset-oleh-apjii-dan-puskakom-ui.

[Bawaslu] Badan Pengawas Pemilihan Umum. 2015. Laporan Kinerja Badan Pengawasan Pemilihan Umum 2014 [Internet]. [diunduh 2015 Desember 5]. Tersedia pada: http://www.bawaslu.go.id/sites/defult/files/lap%20kinerja%20

bawaslu%202014.pdf

[Kominfo] Kantor Komunikasi dan Infomatika Kota Bogor. 2013. Penerapan Teknologi Informasi dan Komunikasi untuk Mendukung Pengembangan *E-Government* Pemerintah Kota Bogor [Internet]. [diunduh 2015 Desember 12]. Tersedia pada: http://kominfo.kotabogor.go.id/asset/file/sop/penerapan-tik-mendukung-pengembangan-e-gov-2014---2018.pdf

Mohammadi S, Nikkahan B. Sohrabi S. 2009. Challenges of user involvement in extreme programming projects. *International Journal of Software Engineering and Its Applications*. 3(1): 20.

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

Ramadhan DA, Nurhardyani Y, Hermadi I. 2015. Analisis dan pengembangan *prototype* aplikasi *web* 2.0 dan media sosial pada kampanye pemilu 2014. Di dalam: *Seminar Nasional Teknologi lnformasi dan Multimedia*; 2015 Februari 6-8; Yogyakarta, Indonesia. Yogyakarta (ID): ISSN. hlm 73−78

[RI] Republik Indonesia. 2007. Undang-Undang Republik Indonesia Nomor 22 Tahun 2007 tentang Penyelenggaraan Pemilu. [Internet]. [diunduh 2016 Juli 21]. Tersedia pada: http://bawaslu.go.id/sites/default/files/regulasi/uu\_22\_

2007.pdf

[RI] Republik Indonesia. 2012. Peraturan Badan Pengawasan Pemilihan Umum Nomor 14 Tahun 2012 tentang Tata Cara Pelaporan Dan Penanganan Pelanggaran Pemilihan Umum Anggota Dewan Perwakilan Rakyat, Dewan Perwakilan Daerah, dan Dewan Perwakilan Rakyat Daerah [Internet]. [diunduh 2016 Agustus 1]. Tersedia pada: http://www.rumahpemilu.com/public/doc/ 2013\_02\_12\_01\_03\_22\_Perbawaslu%20Nomor%2014%20Tahun%202012%20ttg%20Tata%20Cara%20Pelaporan%20dan%20Penanganan%20Pelanggaran%20Pileg.pdf

[RI] Republik Indonesia. 2015. Undang-Undang Republik Indonesia Nomor 1 Tahun 2015 tentang Penetapan Peraturan Pemerintah Pengganti Undang-Undang Nomor 1 Tahun 2014 tentang Pemilihan Gubernur, Bupati, dan Walikota Menjadi Undang-Undang. [Internet]. [diunduh 2016 Juli 11]. Tersedia pada: http://www.dpr.go.id/dokjdih/document/uu/1636.pdf

Lampiran 1 Data laporan pelanggaran

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| judul\_  laporan | tanggal\_  kejadian | Alamat | keterangan | kategori | status |
| Caleg Ir. Sufmi Dasco Ahmad caleg DPR RI dapil Banten III nomor urut 1 | 8-Apr-14 | Jalan Doktor Setia Budi, Pamulang, South Tangerang City, Banten 15417 | Caleg Ir. Sufmi Dasco Ahmad caleg DPR RI dapil Banten III nomor urut 1 telah melakukan pelanggaran dengan memasang alat peraga bersama pada sebuah tiang telpon pada masa tenang di jalan Doktor Setia Budi, Pamulang | Dugaan Pelanggaran Pidana | Verified |
| Calon legislatif DPR RI dapil Banten III bernama Wanto Sugito | 8-Apr-14 | tiang listrik dan tiang telepon sepanjang Jalan Cirendeu Raya | Telah ditemukan dugaan pelanggaran di minggu tenang pada 8 April 2014 pukul 16.00 wib yang di lakukan oleh calon legislatif DPR RI dapil Banten III bernama Wanto Sugito. Temuan pelanggaran tersebut berupa poster di sejumlah tiang listrik dan tiang telepon sepanjang Jalan Cirendeu Raya | Dugaan Pelanggaran Pidana | Verified |

Lampiran 2 Data calon legislatif

25

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Nama | Alamat | Lembaga | Nama Partai | No Urut | Jenis Kelamin | Agama | Tanggal Lahir | Tempat Lahir | Dapil | Pendidikan Terakhir |
| 1 | R. Syaripudin | Cianjur, Jawa Barat | DPR | Partai Persatuan Pembangunan | 5 | L | Islam | 25-Oct-1962 | CIANJUR | JAWA BARAT III | SMAN |
| 2 | Dede Yusuf Macan Effendi, ST | Lebak Gede, Jawa Barat | DPR | Partai Demokrat | 1 | L | Islam | 14-Sep-1966 | Jakarta | JAWA BARAT II | S1 |
| 3 | Aryo P.S. Djojhadikusumo | Karet Tengsin, Jakarta Pusat | DPR | Partai Gerakan Indonesia Raya | 1 | L | Kristen | 25-Apr-1983 | Jakarta | DKI JAKARTA III | - |
| 4 | Ivan Doly Gultom | Penjaringan, Jakarta Utara | DPR | Partai Golongan Karya | 5 | L | Protestan | 3-Apr-1967 | Binjai | DKI JAKARTA III | S1 |
| 5 | Puan Maharani | Kebagusan, Jakarta Selatan | DPR | Partai Demokrasi Indonesia Perjuangan | 1 | P | Islam | 9-Jun-1973 | Jakarta | JAWA TENGAH V | S1 |
| 6 | Syarif, M.Si | Jakarta Timur | DPRD | Partai Gerakan Indonesia Raya | 4 | L | - | - | - | DKI JAKARTA VI | - |
| 7 | Ir. Togu Pardamean Tobing, M.Si. | Bencongan, Banten | DPD | - | 26 | L | Kristen | 11-May-1960 | Sabang | BANTEN | S3 |

Lampiran 3 *Fully developed use case description* iterasi pertama melihat data laporan pelanggaran

|  |  |
| --- | --- |
| *Use case Name:* | Melihat data laporan pelanggaran |
| *Scenario:* | Masyarakat ingin melihat data laporan pelanggaran |
| *Triggering Event:* | Masyarakat memilih menu Pantau lalu dan Data Laporan |
| *Brief Description:* | Melihat data-data laporan yang sebelumnya telah diinputkan oleh masyarakat |
| *Actors:* | Masyarakat |
| *Related Use Case:* | *Login* masyarakat |
| *Stakeholders:* | Masyarakat |
| *Preconditions:* | Data laporan pelanggaran telah diinputkan oleh masyarakat |
| *Postconditions:* | Masyarakat berhasil melihat data laporan pelanggaran |
| *Flow of Activities:* | |  |  | | --- | --- | | *Actor* | *System* | | 1. Memilih menu Pantau 2. Memilih menu Data Laporan | 2.1 Menampilkan halaman menu data laporan | |
| *Exception Conditions:* | * Jika masyarakat belum ada yang menginputkan laporan, maka tidak ada data yang ditampilkan * Jika Bawaslu belum memberikan tanggapan terhadap laporan, maka tanggapan adalah 0 atau tidak ada |

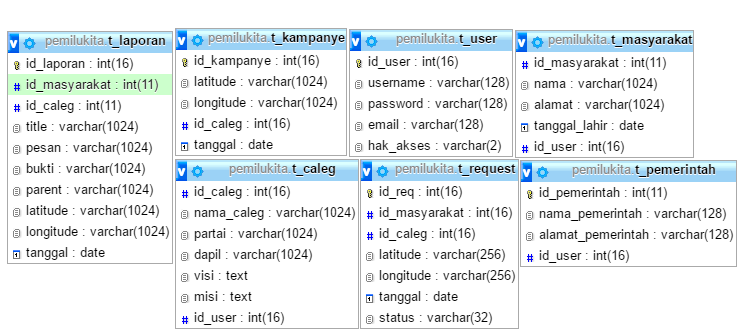
Lampiran 4 *Fully developed use case description* iterasi pertama melakukan *request* lokasi kampanye

|  |  |
| --- | --- |
| *Use case Name:* | Melakukan *request* lokasi kampanye |
| *Scenario:* | Masyarakat ingin diadakan kampanye di lokasi yang diinginkan |
| *Triggering Event:* | Masyarakat memilih menu Kampanye lalu dan Request |
| *Brief Description:* | Me-*request* lokasi kampanye |
| *Actors:* | Masyarakat |
| *Related Use Case:* | *Login* masyarakat |
| *Stakeholders:* | Masyarakat |
| *Preconditions:* | Form *input request* kampanye telah ditampilkan |
| *Postconditions:* | Masyarakat berhasil me-*request* lokasi kampanye |
| *Flow of Activities:* | |  |  | | --- | --- | | *Actor* | *System* | | 1. Memilih menu Kampanye 2. Memilih menu Request 3. Mengisi form *input request* kampanye 4. Pilih ‘Simpan’ pada halaman sistem | 2.1 Menampilkan halaman form Request  4.1 Data *request* kampanye telah masuk ke dalam API Request Kampanye | |
| *Exception Conditions:* | Tidak ada kondisi |

Lampiran 5 *Fully developed use case description* iterasi pertama memberikan tindakan terhadap laporan pelanggaran

|  |  |
| --- | --- |
| *Use case Name:* | Memberikan tindakan terhadap laporan pelanggaran |
| *Scenario:* | Bawaslu ingin memberikan tindakan terhadap laporan pelanggaran yang telah dilaporkan oleh masyarakat |
| *Triggering Event:* | Bawaslu memilih menu Laporan lalu dan Beri Tanggapan pada laporan yang ingin ditanggapi. |
| *Brief Description:* | Memberikan tindakan terhadap laporan masyarakat |
| *Actors:* | Bawaslu |
| *Related Use Case:* | *Login* Bawaslu |
| *Stakeholders:* | Bawaslu |
| *Preconditions:* | Form *input* tanggapan telah ditampilkan |
| *Postconditions:* | Bawaslu berhasil memberikan tanggapan |
| *Flow of Activities:* | |  |  | | --- | --- | | *Actor* | *System* | | 1. Memilih menu Laporan 2. Memilih *button* Beri Tanggapan pada laporan yang akan ditanggapi 3. Mengisi form 4. Pilih ‘Simpan’ pada halaman sistem | 2.1 Menampilkan halaman form Beri Tanggapan  4.1 Data tanggapan telah masuk ke dalam API Laporan | |
| *Exception Conditions:* | Tidak ada kondisi |

Lampiran 6 Rancangan basis datadengan struktur MySQL



Lampiran 7 *Fully developed use case description* iterasi kedua melihat data laporan pelanggaran

|  |  |
| --- | --- |
| *Use case Name:* | Melihat data laporan pelanggaran |
| *Scenario:* | Masyarakat ingin melihat data laporan pelanggaran |
| *Triggering Event:* | Masyarakat memilih menu Laporan |
| *Brief Description:* | Melihat data-data pelanggaran yang sebelumnya telah dilaporkan oleh masyarakat |
| *Actors:* | Masyarakat |
| *Related Use Case:* | *Login* masyarakat |
| *Stakeholders:* | Masyarakat |
| *Preconditions:* | Data laporan pelanggaran telah dilaporkan oleh masyarakat |
| *Postconditions:* | Masyarakat berhasil melihat data laporan pelanggaran |
| *Flow of Activities:* | |  |  | | --- | --- | | *Actor* | *System* | | 1. Memilih menu Laporan | 1.1 Menampilkan halaman menu Laporan | |
| *Exception Conditions:* | * Server telah diaktifkan * Jika masyarakat belum ada yang melaporkan pelanggaran, maka tidak ada data yang ditampilkan * Jika Bawaslu belum memberikan tanggapan terhadap laporan, maka tanggapan berisi kalimat belum ada tanggapan. |

Lampiran 8 *Fully developed use case description* iterasi kedua melaporkan pelanggaran

|  |  |
| --- | --- |
| *Use case Name:* | Melaporkan pelanggaran |
| *Scenario:* | Masyarakat ingin melaporkan pelanggaran |
| *Triggering Event:* | Masyarakat memilih menu Laporan lalu memilih *button* Buat Laporan yang terdapat pada halaman Laporan |
| *Brief Description:* | Melaporkan pelangaran-pelanggaran yang dilihat saat kampanye berlangsung |
| *Actors:* | Masyarakat |
| *Related Use Case:* | *Login* masyarakat |
| *Stakeholders:* | Masyarakat |
| *Preconditions:* | *Form* Buat Laporan sudah ditampilkan |
| *Postconditions:* | Masyarakat berhasil melaporkan pelanggaran |
| *Flow of Activities:* | |  |  | | --- | --- | | *Actor* | *System* | | 1. Memilih menu Laporan 2. Memilih *button* Buat Laporan 3. Mengisi *form* Buat Laporan 4. Pilih ‘*Save*’ pada halaman sistem | 1.1 Menampilkan halaman menu Laporan  2.1 Menampilkan *form* Buat Laporan  4.1 Data laporan pelanggaran telah masuk ke dalam API Laporan | |
| *Exception Conditions:* | * Server telah diaktifkan |

Lampiran 9 *Fully developed use case description* iterasi kedua melakukan *request* lokasi kampanye

|  |  |
| --- | --- |
| *Use case Name:* | Melakukan *request* lokasi kampanye |
| *Scenario:* | Masyarakat ingin me-*request* lokasi kampanye |
| *Triggering Event:* | Masyarakat memilih menu Kampanye lalu memilih *button* Request Kampanye yang terdapat pada halaman Kampanye |
| *Brief Description:* | Melakukan *request* lokasi kampanye kepada Calon Legislatif |
| *Actors:* | Masyarakat |
| *Related Use Case:* | *Login* masyarakat |
| *Stakeholders:* | Masyarakat |
| *Preconditions:* | *Form* Request Kampanye sudah ditampilkan |
| *Postconditions:* | Masyarakat berhasil me-*request* lokasi kampanye |
| *Flow of Activities:* | |  |  | | --- | --- | | *Actor* | *System* | | 1. Memilih menu Kampanye 2. Memilih *button* Request Kampanye 3. Mengisi *form* Request Kampanye 4. Pilih ‘*Save*’ pada halaman sistem | 1.1 Menampilkan halaman menu Kampanye  2.1 Menampilkan *form* Request Kampanye  4.1 Data *request* kampanye telah masuk ke dalam API Request Kampanye | |
| *Exception Conditions:* | * Server telah diaktifkan |

Lampiran 10 *Fully developed use case description* iterasi kedua memberikan tindakan terhadap laporan pelanggaran

|  |  |
| --- | --- |
| *Use case Name:* | Memberikan tindakan terhadap laporan pelanggaran |
| *Scenario:* | Bawaslu ingin memberikan tindakan terhadap laporan pelanggaran yang telah dilaporkan oleh masyarakat |
| *Triggering Event:* | Bawaslu memilih menu Laporan lalu memilih *button* Belum Respon yang terdapat pada halaman Kampanye kemudian memilih laporan yang akan diberikan tindakan |
| *Brief Description:* | Memberikan tindakan terhadap laporan pelanggaran |
| *Actors:* | Bawaslu |
| *Related Use Case:* | *Login* Bawaslu |
| *Stakeholders:* | Bawaslu |
| *Preconditions:* | *Form* Beri Tindakan sudah ditampilkan |
| *Postconditions:* | Bawaslu berhasil memberikan tindakan terhadap laporan pelanggaran |
| *Flow of Activities:* | |  |  | | --- | --- | | *Actor* | *System* | | 1. Memilih menu Laporan 2. Memilih *button* Belum Respon 3. Memilih laporan yang akan diberikan tindakan dan meng-*klik* *button* Beri Tanggapan 4. Pilih ‘*Save*’ pada halaman sistem | 1.1 Menampilkan halaman menu Laporan  2.1 Menampilkan data laporan yang belum direspon  3.1 Menampilkan *form* Beri Tindakan  4.1 Data tanggapan telah masuk ke dalam API Laporan | |
| *Exception Conditions:* | * Server telah diaktifkan * Masyarakat telah melaporkan pelanggaran * Terdapat laporan pelanggaran yang belum ditanggapi |

Lampiran 11 *Swimlane diagram task* melihat data laporan pelanggaran dan melaporkan pelanggaran



Lampiran 12 *Swimlane diagram task* melakukan *request* lokasi kampanye



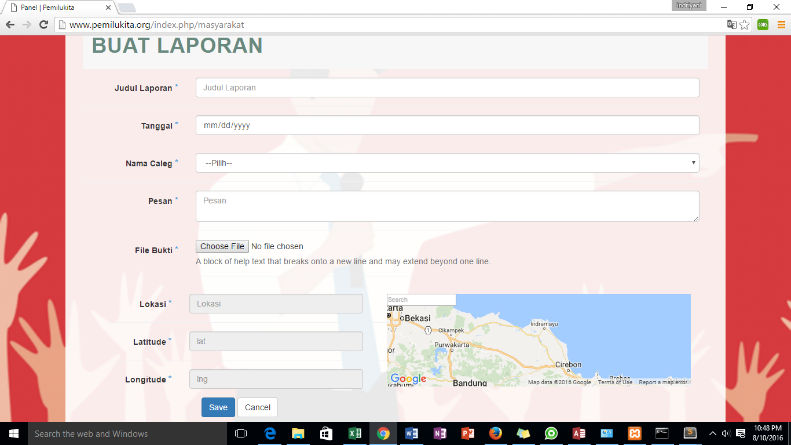
Lampiran 13 Rancangan basis datadengan struktur NoSQL

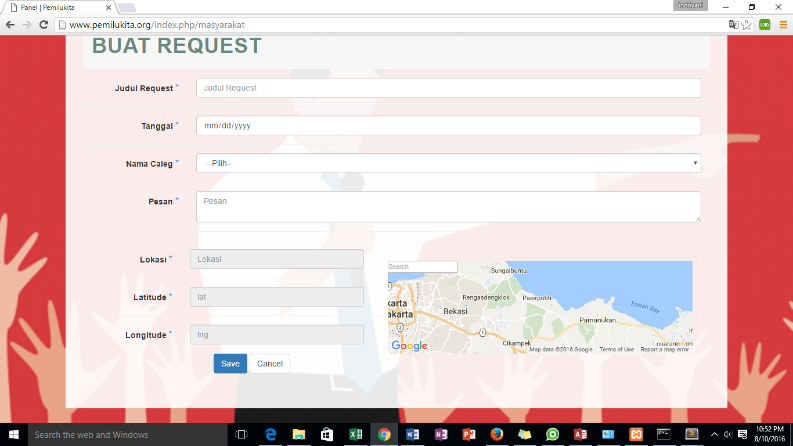
Lampiran 14 Implementasi pada *web* *browser*

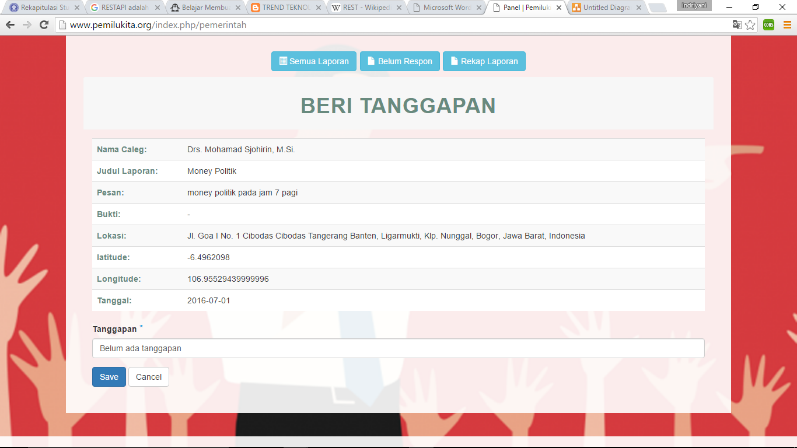
1. Halaman melihat data laporan pelanggaran



1. Halaman melaporkan pelanggaran



1. Halaman melakukan *request* lokasi kampanye
2. Halaman memberikan tindakan terhadap laporan pelanggaan



**RIWAYAT HIDUP**

Penulis dilahirkan di Jakarta pada tanggal 3 Mei 1994 dari Ayah Yan B. Bastian dan Ibu Neni. Penulis adalah putri kelima dari lima. Penulis menyelesaikan pendidikan MTs Negeri 7 Model Jakarta pada tahun 2009 dan tahun 2012 penulis lulus dari SMA Negeri 105 Jakarta dan pada tahun yang sama penulis lulus seleksi masuk Institut Pertanian Bogor (IPB) melalui jalur SNMPTN Undangan dan diterima di Departemen Ilmu Komputer. Pada 29 Juni Akhir – 21 Agustus 2015 penulis melaksanakan praktik kerja lapangan di Kantor Komunikasi dan Informatika (Kominfo) Kota Bogor dengan bidang kajian Webgis Kota Bogor – OPD dan Kelurahan Berbasis Titik.