

PERANCANGAN SISTEM INFORMASI PENCATATAN METERAN AIR PDAM BERBASIS WEB MENGGUNAKAN FRAMEWORK MVC STUDI KASUS PDAM BOGOR

Sirojul Munir, Indra Hermawan
STT Terpadu Nurul Fikri, Depok – Jawa Barat
rojulman@nurulfikri.ac.id, indra@nurulfikri.ac.id,

Pemanfaatan teknologi informasi menjadi salah satu kebutuhan penting dari institusi, terutama institusi publik yang melayani pelanggan seperti Perusahaan Daerah Air Minum (PDAM). Sistem informasi pencatatan meteran pelanggan dibutuhkan untuk memberikan akses informasi kepada pihak yang berkepentingan dalam hal ini manajemen PDAM dan pelanggan. Penelitian ini berfokus pada perancangan sistem informasi pencatatan meteran PDAM berbasis web pada institusi PDAM kabupaten Bogor, yang meliputi modul administrasi pencatatan meteran pelanggan dan modul penyediaan layanan web (*web service*). Metode *unified process* digunakan pada perancangan sistem, dan framework web dengan arsitektur desain *Model View Controller (MVC)* digunakan untuk membangun aplikasi. Dari penelitian diharapkan dihasilkan prototype aplikasi yang dapat digunakan untuk pengelolaan meteran pelanggan serta tersedianya *web service* yang dapat diakses oleh platform aplikasi lain.

Kata Kunci: Pencatatan meteran PDAM, web framework, MVC, *web service*

1. PENDAHULUAN

Pada era teknologi informasi kebutuhan akses ke sumber data menjadi suatu kebutuhan penting bagi sebuah institusi publik dalam melayani pelanggannya. Jumlah pelanggan yang banyak beserta transaksi yang dilakukan membutuhkan pengelolaan yang tidak mudah, apalagi dilakukan secara manual atau bahkan menggunakan komputer sekalipun.

Perusahaan Daerah Air Minum (PDAM) Bogor melayani pelanggannya dalam pengelolaan air bersih. Pelanggan air PDAM saat ini dalam akses data terhadap transaksi yang sudah dilakukan terbatas hanya sebagai kuitansi pembayaran, demikian juga pencatatan pelaporan transaksi penggunaan, di sisi lain pencatatan meteran pelanggan PDAM oleh petugas lapangan dilakukan secara manual untuk kemudian disimpan dalam sistem komputer, hal ini sangatlah tidak efisien karena melewati beberapa tahapan yang jika tidak dilakukan dengan teliti maka akan cenderung terjadi kesalahan.

Sistem informasi pencatatan meteran pelanggan dibutuhkan oleh PDAM Bogor untuk proses administrasi dan pengelolaan data meteran pelanggan yang telah dikumpulkan oleh operator pencatat meteran di lapangan, agar proses pencatatan meteran PDAM efisien dan akurat maka dibutuhkan layanan web (*web service*) yang dapat diakses oleh aplikasi pencatatan meteran milik operator lapangan menggunakan aplikasi *smartphone*. Ketersediaan *web service* dapat digunakan untuk mengambil informasi detail pelanggan dan menyimpan hasil pencatatan meteran pelanggan. Bagi kepentingan akses data ke pelanggan *web service* dapat pula digunakan untuk mendapatkan informasi transaksi penggunaan kubik air serta informasi pembayaran atau

tagihan bulanan pemakaian melalui aplikasi web ataupun aplikasi mobile.

Fokus penelitian ini adalah bagaimana merancang sistem informasi pencatatan meteran air PDAM berbasis web serta penyediaan *web service* agar dapat diakses oleh pihak yang berkepentingan seperti manajemen PDAM dan pelanggan.

Tujuan umum dikembangkannya sistem informasi pencatatan meteran air PDAM ini adalah untuk efisiensi akses data oleh pengguna. Sedangkan tujuan khusus perancangan sistem informasi pencatatan meteran air PDAM ini adalah untuk mengetahui implementasi penggunaan web framework berbasis Model View Controller (MVC) dalam membangun sistem informasi berupa prototipe aplikasi web pencatatan meteran air PDAM beserta *web service*-nya, dengan fokus pada *use cases* yang sesuai dengan *requirement* dari sistem yang dikembangkan, yaitu:

- Mendukung kemudahan akses terhadap administrasi data meteran pelanggan oleh manajemen
- Mendukung kemudahan akses data web service oleh aplikasi pencatatan meteran pelanggan
- Mendukung kemudahan akses data web service oleh aplikasi yang dimiliki pelanggan

2. TEORI DASAR

2.1 Sistem Informasi

Menurut Robert A. Leitch dan K. Roscoe Davis “sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan

laporan-laporan yang diperlukan” [1].

Sistem informasi memiliki komponen-komponen pembentuk, menurut John Burch dan Gary Grudnitski Sistem Informasi terdiri dari komponen-komponen yang disebut dengan istilah blok *bangunan* (*building block*), yaitu blok masukan (*input block*), blok model (*model block*), blok keluaran (*output block*), blok teknologi (*technology blok*), blok basis data (*database block*) dan blok kendali (*controls block*). Keenam blok tersebut harus saling berinteraksi satu sama lain untuk mencapai sasaran dalam satu kesatuan [1].

2.2 Web Framework MVC

Aplikasi web adalah aplikasi *client-server* yang menggunakan *browser* web sebagai klien. *Browser* mengirim permintaan ke server, dan server menghasilkan tanggapan dan mengembalikan hasil ke browser [2].

Sebuah aplikasi web sederhana memiliki tiga tingkatan logis dasar atau lapisan: presentasi, logika bisnis, dan data. Setiap lapisan dapat didefinisikan sebagian bagian abstraksi yang berbeda dari sebuah arsitektur. Semua lapisan dapat berjalan di server aplikasi yang sama pada mesin tunggal, atau pada tiga atau lebih server aplikasi pada mesin terpisah [3].

Dalam bidang pengembangan perangkat lunak, framework adalah sebuah kerangka kerja yang menyediakan kumpulan file-file pustaka program (*libraries*) dan atau berupa class-class yang ketika digunakan akan bekerja sama untuk mengerjakan suatu perintah tertentu dalam mencapai tujuan atau keinginan suatu domain pekerjaan, sedangkan aplikasi framework adalah sebuah kerangka-kerangka (*skeletons*) dari himpunan aplikasi-aplikasi yang bisa dikustomisasi oleh pengembang aplikasi (programmer) [4].

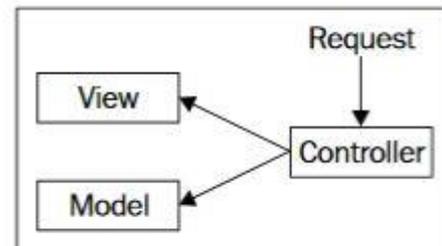
Framework-framework aplikasi *object oriented* adalah suatu solusi untuk membangun aplikasi yang berkualitas pada suatu domain pekerjaan dimana secara sistematis menggunakan desain secara berulang-ulang [5]. Menurut Schwabe [4], sebuah framework aplikasi *object oriented* adalah sebuah desain yang dapat digunakan ulang (*re-usable*) yang dibangun dari kumpulan class-class dan model dari object yang saling bekerjasama.

Pola desain pengembangan perangkat lunak (design pattern) merupakan deskripsi dari class-class dan object-object yang saling berkomunikasi yang tersusun untuk memecahkan masalah perancangan secara umum pada sebuah konteks tertentu [6].

Pola desain *Model View Controller* (*MVC*) terdiri atas 3 jenis objek, *Model* adalah objek aplikasi, *View* adalah objek antar muka tampilan presentasi dan *Controller* adalah bagaimana reaksi pengguna saat menerima inputan [6].

Sekitar awal tahun 2005, pola desain Model-View-Controller untuk aplikasi berbasis web mulai diperkenalkan. Pola desain pengembangan perangkat lunak ini memisahkan antara data (model), user

antarmuka (*view*), dan penanganan data logika (*controller*) [7]. Beberapa manfaat dari pola desain MVC adalah: perubahan kode program oleh user dapat dilakukan tanpa mempengaruhi yang lain, seorang desainer dapat bekerja pada antarmuka tanpa khawatir tentang penyimpanan dan pengelolaan data, bagi pengembang dapat memprogram penanganan logika tanpa masuk kedalam rincian presentasi [7].



Gambar 1 Pola desain Model-View-Controller [6]

Yii adalah web framework berbasis pemrograman PHP yang menggunakan pendekatan Object Oriented Programming [8]. Yii framework adalah framework berbasis komponen yang dapat digunakan untuk mengembangkan aplikasi web pada skala apapun, sehingga Yii framework dapat mendorong penggunaan kembali kode maksimum pemrograman web dan secara signifikan dapat mempercepat proses pembangunan aplikasi [8].

Yii juga menjanjikan aplikasi yang dibuat akan aman (*secure*), karena Yii menerapkan fitur input validasi, filter output, dan pencegahan dari *SQL injection* dan *Cross-site scripting*. Selain itu Yii adalah framework yang profesional karena Yii dapat membantu anda dalam pengembangan aplikasi dengan kode program yang bersih dan dapat digunakan kembali (*reusable*), aplikasi yang dibuat dengan Yii menggunakan pendekatan pola desain MVC yang memastikan pemisahan antara lapisan logika dan lapisan presentasi (antar muka user) [8].

2.3 Web Service

Menurut David Chappel dan Tyler Jewell, *web service* adalah bagian dari logika bisnis, terletak di suatu tempat di Internet, yang dapat diakses melalui protokol internet berbasis standar seperti HTTP atau SMTP [9].

Menurut Arora dan Kishore, *web service* adalah sebuah teknologi yang mengizinkan untuk membuat aplikasi yang independen terhadap platform pengembangan. *Web service* dapat dikembangkan dengan beragam bahasa dan beragam platform yang mengikuti standar teknologi. Sebuah *web service* merupakan suatu aplikasi yang membuka kode fungsionalitas aplikasi ke beberapa aplikasi [10].

Web service dibangun atas empat teknologi standar berikut ini [11]:

- *Extensible Markup Language (XML)*, adalah

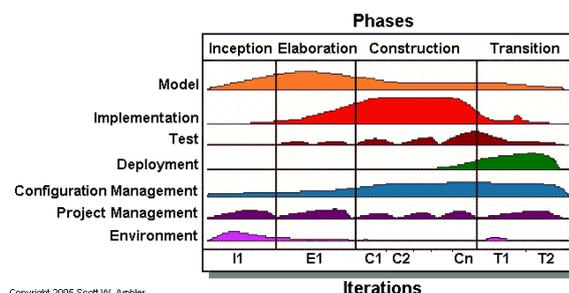
lapisan representasi data untuk semua protokol web service dan teknologi yang dibuat didalamnya

- **Simple Object Access Protocol (SOAP)**, dirancang untuk menjadi: sederhana dan sebagai spesifikasi extensible untuk pertukaran informasi yang terstruktur, berbasis XML pada proses desentralisasi, didistribusikan lingkungan.
- **Web Services Description Language (WSDL)**, sebagai standar untuk menjelaskan web service sebagai layanan abstrak pada operasi pesan yang berbeda format dan berbeda protokol jaringan.
- **Universal Description, Discovery, and Integration Standard (UDDI)**, sebagai alat standar yang digunakan untuk menerbitkan dan menemukan web service.

REST web service merupakan web service yang dibangun dengan memanfaatkan teknologi dan protokol yang sudah ada. Menurut Roy Thomas Fielding, REST (REpresentational State Transfer) adalah model arsitektur yang pada dasarnya memanfaatkan teknologi dan protokol yang sudah ada seperti HTTP (Hypertext Transfer Protocol) dan XML. [12]

2.4 Unified Process

Menurut Carg Larman, *Unified Process* (UP) merupakan suatu metode pembangunan perangkat lunak yang menggunakan sistem *object oriented*. *Unified Process* menggunakan proses iteratif pada proyek yang menggunakan pendekatan desain dan analisa objek oriented (OOAD) [13].



Copyright 2005 Scott W. Ambler

Gambar 2. Proses iterasi Unified Process

Dalam *unified process* terdapat beberapa fase yang dilakukan secara iteratif [13], yaitu:

- **Inception:** fase awal menangkap visi yang diinginkan, memahami bisnis proses, menentukan skop pekerjaan, estimasi dan kelayakan dari proyek.
- **Elaboration:** fase mendefinisikan ulang visi yang diinginkan, iterasi implementasi arsitektur inti, menentukan faktor resiko proyek, identifikasi requirement utama dan skop pekerjaan, didapat waktu realistis pengerjaan proyek
- **Construction:** pada fase dilakukan iterasi implementasi dan mengerjakan elemen tersisa yang beresiko rendah dan mudah, dilakukan proses

persiapan deployment.

- **Transition:** fase testing dan transisi ke deployment.

UML (*Unified Modelling Language*) merupakan alat bantu yang memiliki kemampuan handal di dalam pengembangan sistem yang berorientasi objek. Kehandalan UML ini dibuktikan dengan disediakannya bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk baku, mudah dimengerti, dan dilengkapi dengan mekanisme yang efektif untuk berbagi serta mengkomunikasikan rancangan mereka dengan yang lain [14].

UML terbuat dari beberapa model yang menjelaskan bagaimana sistem di lakukan desain, model-model digambarkan dalam bentuk diagram-diaagram: Use Case Diagram, Class Diagram, Object Diagram, Activity Diagram, Sequence Diagram, Collaboration Diagram, Statechart Diagram, Component Diagram dan Deployment Diagram [13].

2.5 Black Box Testing

Pengujian dengan metode *black-box* juga biasa disebut dengan behavioral testing yang berfokus pada functional requirement dari sebuah perangkat lunak. Metode pengujian *black-box* ini memungkinkan seorang software engineer dapat memberikan sekumpulan input guna menguji semua fungsionalitas requirement dari sebuah program. *Black-box* testing ini merupakan pelengkap dari tahapan pengujian sebuah software selain metode *white box*. Metode *black-box* ini tidak seperti *white box* yang dapat dilakukan di awal tetapi pengujian *black-box* dilakukan pada tahap akhir, artinya sebelum program secara keseluruhan selesai program dapat diuji dengan metode *white-box* untuk menguji control structure dari sebuah program sedangkan *black-box* mengabaikan control structure dan lebih berfokus pada domain informasi apakah input yang diberikan pengujian menghasilkan keluaran yang diharapkan oleh user. Pada pengujian *black-box* kesalahan yang berusaha ditemukan antara lain [15]:

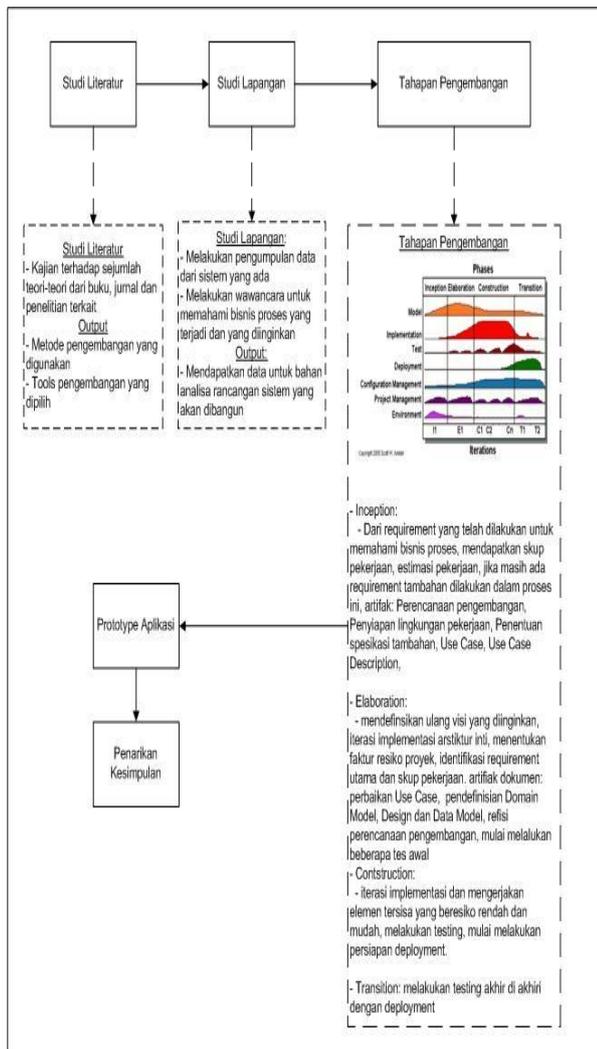
- Kesalahan Performa
- Kesalahan interface
- Kesalahan dalam struktur data atau akses database eksternal
- Fungsi-fungsi yang salah atau hilang

Salah satu metode *black-box* testing yaitu *equivalence class partitioning*, *equivalence class partitioning* adalah metode uji coba *black-box* yang membagi domain input dari program menjadi beberapa kelas data dari kasus uji coba yang dihasilkan. *Equivalence class partitioning* berusaha untuk mendefinisikan kasus uji agar dapat menemukan sejumlah kesalahan, kasus uji yang didesain untuk *equivalence class partitioning* berdasarkan pada evaluasi dari ekuivalensi jenis atau class untuk kondisi

input. *Class-class* yang ekuivalen merepresentasikan sekumpulan keadaan valid dan invalid untuk kondisi input. Kondisi input sendiri dapat berupa nilai numerik yang spesifik, kisaran nilai (range), sekumpulan nilai yang berhubungan (himpunan), atau kondisi boolean [15].

3. KERANGKA PIKIR PENELITIAN

Terdapat tiga tahapan dalam penelitian ini, pertama melakukan studi literatur untuk mendapatkan metode pengembangan yang akan digunakan serta melakukan pemilihan teknologi dan tools yang akan diimplementasikan. Tahap berikutnya adalah melakukan studi lapangan untuk mengumpulkan data, mendefinisikan keinginan atau visi dari stakeholder sehingga didapat bahan untuk analisis rancangan sistem yang akan dibangun, tahap selanjutnya adalah tahapan pengembangan yang meliputi empat fase iterasi menggunakan metode unified process, yaitu: *inception, elaboration, construction* dan *transition*.



Gambar 3. Bagan Kerangka Pikir Penelitian

Proses uji coba prototype aplikasi dilakukan secara iterasi mulai pada tahapan elaboration, construction

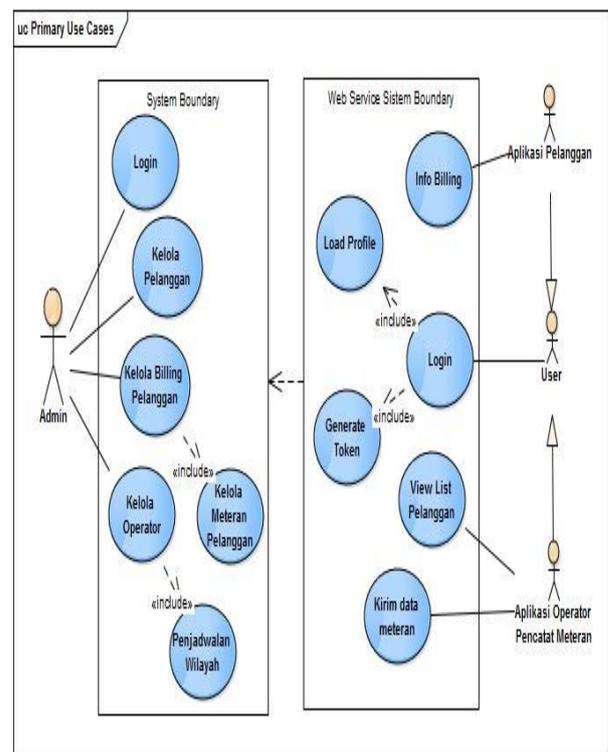
dan transition dengan iterasi mulai dari use case berat hingga yang ringan, dari proses pengembangan diharapkan dihasilkan *prototype* aplikasi untuk kemudian dilakukan uji akhir bersifat pengujian fungsionalitas dengan menggunakan metode *black-box* testing, dan kemudian ditarik kesimpulan dan saran penelitian.

4. PEMBAHASAN DAN ANALISIS

4.1 Perancangan Sistem

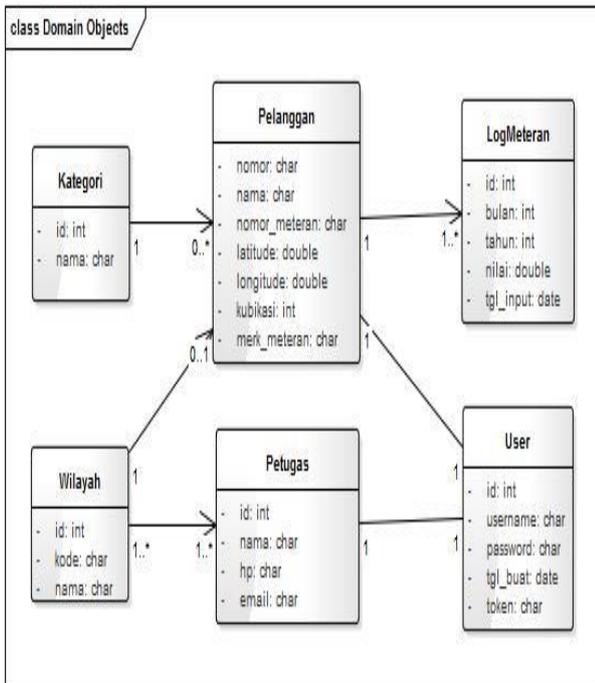
Penelitian ini menggunakan metodologi studi kasus perancangan sistem informasi pencatatan meteran air PDAM pada instansi PDAM kabupaten Bogor. Pengujian hasil berupa prototipe aplikasi yang dikembangkan dengan teknologi berbasis web dengan menggunakan framework web berbasis model-view-controller. Dalam proses analisa yang didapat dari requirement diidentifikasi dua proses bisnis utama yaitu, proses bisnis pengelolaan data meteran pelanggan dan tersedianya *web service* yang dapat diakses oleh aplikasi pencatatan meteran operator dan aplikasi yang dimiliki pelanggan.

Dari use case yang telah didefinisikan diidentifikasi dua kelompok use case, yaitu use case pengelolaan data administrasi meteran pelanggan dan use case untuk akses layanan web service oleh aplikasi operator atau pelanggan tergambar dalam gambar 4



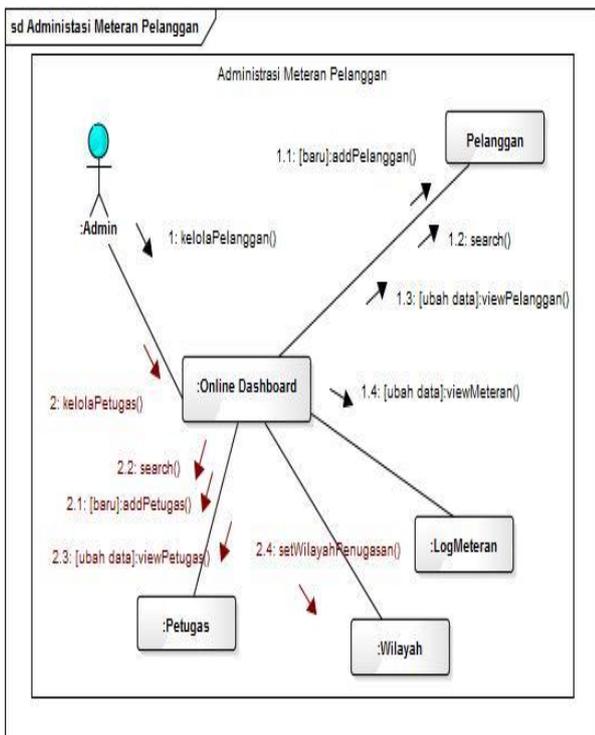
Gambar 4. Diagram Use case

Dari use case didefinisikan domain model dengan entitas utama: pelanggan, kategori, logmeteran, wilayah, petugas dan user, terlihat pada gambar 5.

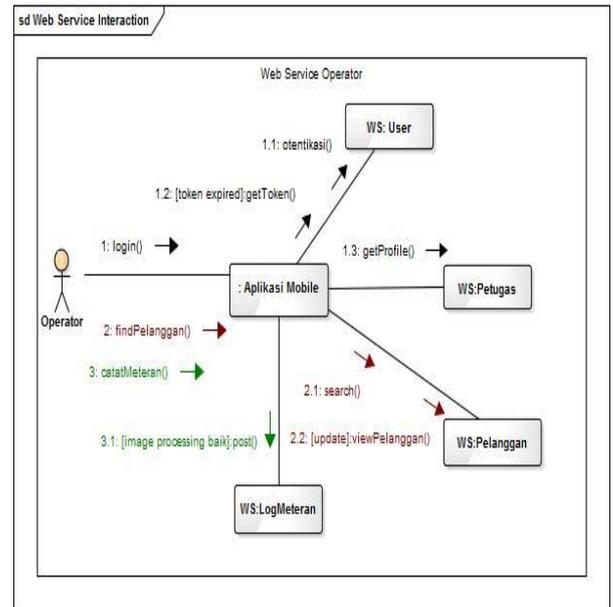


Gambar 5. Diagram Object Model

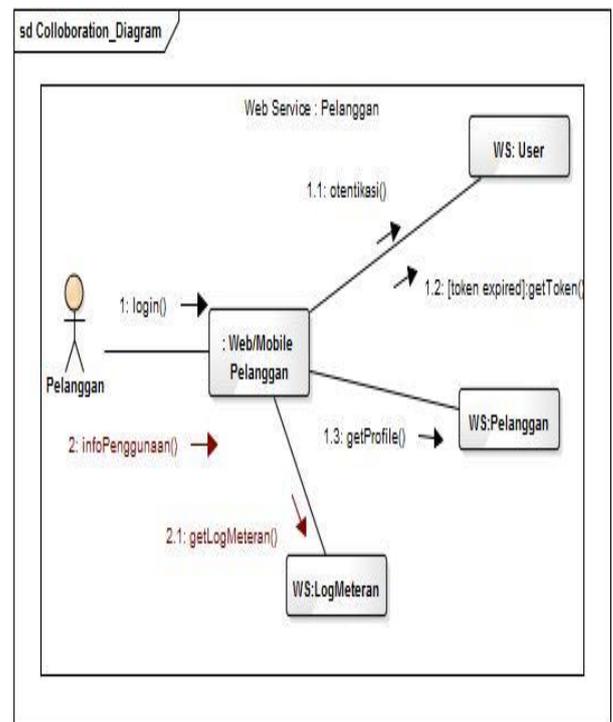
Alur proses data digambarkan dalam diagram collaboration: administrasi meteran pelanggan (gambar 6), web service aplikasi operator (gambar 7) dan web service akses aplikasi pelanggan (gambar 8).



Gambar 6. Diagram collaboration administrasi meteran pelanggan



Gambar 7. Diagram Collaboration Akses Web Service oleh Aplikasi Operator



Gambar 8. Diagram Collaboration Akses Web Service oleh Aplikasi Pelanggan

4.2 Pemilihan Teknologi

Implementasi pengembangan penelitian ini menggunakan framework web model-view-controller Yii2 Framework, yaitu framework web berbasis komponen bahasa pemrograman PHP5. Yii framework bekerja dengan membantu pengembang aplikasi berbasis web dengan memastikan kode program yang dibuat efisien, dapat dikembangkan (extensible), serta memudahkan maintenance produk yang dihasilkan.

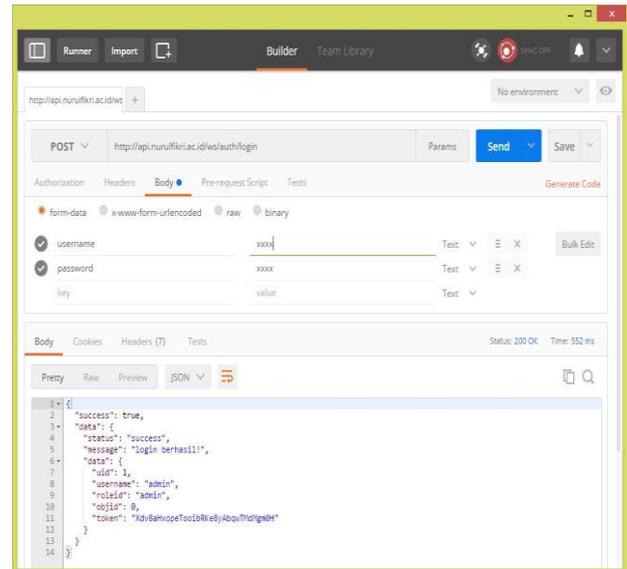
Yii Framework adalah framework web open source

yang kode programnya dibuat oleh Qiang Xue diawal Januari 2008 yang dapat digunakan untuk pengembangan aplikasi berbasis web 2.0. Pengembangan aplikasi web menggunakan Yii menjanjikan aplikasi yang dapat dibuat dengan cepat (*fast*) karena Yii hanya akan meload fitur-fitur yang dibutuhkan saja, yaitu dengan memanfaatkan dukungan caching, dan Yii didesain untuk bekerja dengan efisien dengan teknologi AJAX [8].

Pada implementasi web service, Yii menyediakan pustaka program (API) RESTful yang dapat digunakan untuk membuat prototype web service secara cepat [16]. Berikut ini fitur-fitur web service yang disupport:

- Penggunaan *ActiveRecord* untuk mempercepat prototype implementasi *web service*
- Mendukung format response dengan data JSON maupun XML;
- Kemudahan kustomisasi Objek serialisasi dengan output pilihan yang fleksibel
- Tersedia formatting koleksi data dan validasinya
- mendukung HATEOAS;
- routing yang efisien dengan fitur HTTP verb check;
- Secara built-in mendukung konsep *verb header OPTIONS* and *HEAD* ;
- Memiliki proses *Authentication* and *authorization*;
- Memiliki *Data caching* dan *HTTP caching*;

Pada proses pengujian *web service* dilakukan dengan menggunakan tools plugin POSTMAN pada web browser yang digunakan.



Gambar 9. Antar muka aplikasi POSTMAN untuk pengujian web service

4.3 Prototype Aplikasi

Hasil prototype aplikasi Sistem Informasi Pencatatan Meteran Air PDAM Berbasis Web Menggunakan Framework MVC Studi Kasus PDAM Bogor yang telah di implementasikan dijelaskan dalam sebuah tabel yang berisikan pemetaan antara tujuan pengembangan sistem dengan use cases, pengujian dilakukan dengan metode *black-box* testing.

Tabel 2 : Hasil uji coba prototipe aplikasi

Tujuan	Use Case / Requirement	Implementasi	Hasil Pengujian
Mendukung kemudahan akses terhadap administrasi data meteran pelanggan oleh manajemen	Login	Ya	Sesuai
	Kelola Pelanggan	Ya	Sesuai
	Kelola Billing Pelanggan	Ya	Sesuai
	Kelola Data Operator	Ya	Sesuai
	Penjadwalan Wilayah	Ya	Sesuai
	Kelola Meteran Pelanggan	Ya	Sesuai
Mendukung kemudahan akses data web service oleh aplikasi pencatatan meteran pelanggan	Login	Ya	Sesuai
	Generate Token	Ya	Sesuai
	Load profile	Ya	Sesuai
	View list pelanggan	Ya	Sesuai
	Kirim Data Pencatatan Meteran	Ya	Sesuai, dengan catatan: disertakan upload gambar
Mendukung kemudahan akses data web service oleh aplikasi yang dimiliki pelanggan	Login	Ya	Sesuai
	Generate Token	Ya	Sesuai
	Load profile	Ya	Sesuai
	Info Billing	Ya	Sesuai

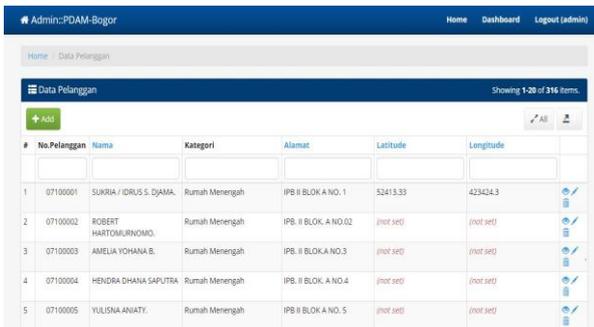
Dapat disimpulkan semua use case telah diimplementasikan pada prototype aplikasi, terdapat catatan pada use case kirim data pencatatan meteran untuk disertakan pula fitur upload gambar menggunakan web service. Berikut contoh tampilan halaman utama aplikasi sistem informasi pencatatan meteran air PDAM.



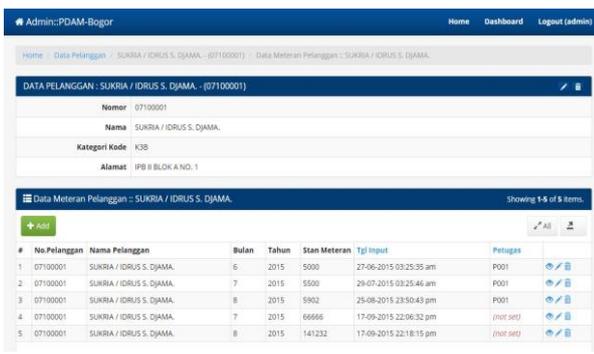
Gambar 10. Halaman depan aplikasi



Gambar 11. Halaman menu dashboard admin



Gambar 12. Halaman kelola data pelanggan



Gambar 13. Halaman pencatatan meteran pelanggan

5. KESIMPULAN

Pada penelitian studi kasus perancangan sistem sistem informasi pencatatan meteran air PDAM studi kasus PDAM Bogor, didapat bahwa aplikasi yang dibangun telah sesuai dengan tujuan pengembangan sistem mendukung kemudahan akses terhadap administrasi data meteran pelanggan oleh manajemen, mendukung kemudahan akses data web service oleh aplikasi pencatatan meteran pelanggan, dan mendukung kemudahan akses data web service oleh aplikasi yang dimiliki pelanggan.

Saran untuk penelitian selanjutnya adalah web service yang telah diimplementasikan pada penelitian ini dapat diujicobakan pada aplikasi client web service sebenarnya, baik melalui aplikasi smartphone maupun web.

6. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Kementerian Riset Teknologi Dan Pendidikan Tinggi Republik Indonesia atas bantuan finansial terhadap penelitian ini. Selain itu penulis juga mengucapkan terima kasih kepada Sekolah Tinggi Teknologi Terpadu Nurul Fikri yang telah membantu dari sisi sarana dan prasarana sehingga penelitian ini dapat terlaksana dengan baik.

7. DAFTAR PUSTAKA

- [1] Jogyanto, "Analisis dan Desain Sistem Informasi", Yogyakarta: Penerbit Andi, 2003.
- [2] L. Shklar, R. Rosen, "Web Application Architecture", John Wiley & Son Ltd, 2003
- [3] N. Dai, L. Mandel, A. Ryman, "eclipse tools platform", Addison-Wesley, 2007.
- [4] Schwabe, Daniel & Rossi, Gustavo Rossi & Esmeraldo, Luiselena dan Lyardet, Fernando. "Web Design Framework: An approach to improve reuse in Web applications", Departamento de Informatica, PUC-Rio, Brazil 2001
- [5] M. Fayad, D. Schmidt and R. Johnson (editors): "Building Application Frameworks", Wiley 1999.
- [6] Erich Gamma, John Vlissides, Ralph Johnson, and Richard Helm "Design Pattern Elements of Reusable Object-Oriented Software", Addison Wesley, 2000
- [7] Ayman Hourieh, "Learning Website Development with Django", Packt Publishing, Birmingham-mumbai 2008
- [8] Jeffrey Winesett, "Web Application Development with Yii and PHP", Packt Publishing, Birmingham-mumbai, 2012
- [9] David Chappel, Tyler Jewell, "Java Web Services", O'Reilly, First Edition 2002
- [10] G. Arora, S. Kishore, "XML Web Services: Professional Projects", Premier Press, 2002

- [11] James O'Grady, Ian Burnett, "Application Development for IBM CICS Web Services", IBM, 2015
- [12] Fielding, R.T. "Architectural Styles and The Design of Networked based Software architectures", Irvine:s.n., 2000. PhD Dissertation
- [13] Carg Larman, "Applying UML And Patterns", Prentice Hall, 2004
- [14] M. Pemodelan Visual Dengan UML, Yogyakarta: Graha Ilmu, 2005.
- [15] R. S. Pressman, "Software Engineering," in *A PRACTITIONER'S APPROACH*, New York, McGraw-Hill, 2001, p. 36.
- [16] Dokumentasi online Yii2 RESTful, <http://www.yiiframework.com/doc-2.0/guide-rest-quick-start.html>, diakses terakhir 18-06-2016

Biodata Penulis



Sirojul Munir, memperoleh gelar Sarjana Matematika (S.Si), Jurusan Matematika Universitas Indonesia Depok, lulus tahun 1996. Memperoleh gelar Magister Komputer (M.Kom) Program Pasca Sarjana Magister Ilmu Komputer Universitas Budi Luhur Jakarta, lulus tahun 2006. Saat ini menjadi Dosen di STT Terpadu Nurul Fikri Depok, Jawa Barat.



Indra Hermawan, memperoleh gelar Sarjana Komputer (S.Kom), Jurusan Teknik Komputer Universitas Komputer Indonesia Bandung, lulus tahun 2009. Memperoleh gelar Magister Komputer (M.Kom) Program Pasca Sarjana Magister Ilmu Komputer Universitas Indonesia

Depok, lulus tahun 2013. Saat ini menjadi Dosen di STT Terpadu Nurul Fikri Depok, Jawa Barat.