

Rancangan Web Service Dengan Metode REST API Untuk Integrasi Aplikasi Mobile dan Website Pada Bank Sampah

Muhammad Widyan Riadhi Fakhrun¹⁾, Soni Fajar Surya Gumilang²⁾

Program Studi S1 Sistem Informasi, Fakultas Rekayasa Industri, Universitas Telkom
Jl. Telekomunikasi No. 1, Terusan Buah Batu, Dayeuhkolot Bandung 40257, +62(22)7564108
e-mail: muhammadwidyan36@gmail.com¹⁾, mustonie@telkomuniversity.ac.id²⁾

Abstrak

Permasalahan sampah adalah salah satu isu yang serius di Indonesia. Pengolahan sampah dengan cara konvensional belum cukup membantu untuk merubah paradigma masyarakat. Adapun pengolahan sampah dengan cara modern yaitu didirikan bank sampah. Bank sampah sudah memiliki gagasan berupa aplikasi untuk menjalankan bisnisnya, dengan membuat aplikasi mobile dan website. Untuk menjadi lebih efektif dan hemat biaya dalam proses bisnis bank sampah perlu untuk mengimplementasikan web service. Metode web service yang digunakan adalah REST API. REST API sangat mudah untuk diimplementasikan dan digunakan untuk pengembangan lebih lanjut. Penelitian ini mengusulkan sistem yang saling terintegrasi antara aplikasi mobile dan website pada bank sampah agar pertukaran informasi dan proses administrasi berjalan dengan lancar dan akurat. Rancangan sistem ini membawa cara baru dalam memanfaatkan teknologi informasi dalam hal bisnis, sosial, dan lingkungan. Hasil dari perancangan sistem ini menghasilkan end point untuk modul yang tersedia pada fitur aplikasi yang ada pada bank sampah dan nasabah. Pada aplikasi bank sampah terdiri dari end point modul kelola sampah, kelola nasabah, kelola jadwal jemput sampah, dan transaksi tabungan. Sedangkan, untuk nasabah terdiri dari end point untuk lihat profil, lihat data sampah, lihat riwayat transaksi, lihat jadwal jemput sampah dan memilih jemput sampah.

Kata kunci: sampah, aplikasi, mobile, website, service

1. Pendahuluan

Permasalahan sampah dari sisi distribusi, pengelolaan dan pengolahan menjadi masalah yang serius saat ini. Permasalahan ini dibuktikan dengan data dari Menteri Lingkungan Hidup bahwa produksi sampah di Indonesia mencapai 64 juta ton per tahun, dan komposisi sampah: organik 60%, plastik 15%, kertas 10%, lainnya (metal, kaca, kain, kulit) 15%. Berdasarkan data kenaikan produksi sampah di Indonesia, dengan begitu permasalahan sampah belum selesai, sehingga perlu ada perhatian serius untuk menanganinya. Jika dibiarkan maka akan muncul dampak serius seperti lingkungan tidak bersih, pencemaran lingkungan dan dampak negatif lainnya.

Pengelolaan sampah memiliki berbagai cara, yaitu pengelolaan sampah secara konvensional dan modern. Secara konvensional kita ketahui ada yang namanya pengepul, dan yang paling baru saat ini adalah bank sampah. Bank sampah adalah lembaga yang bergerak di bidang bisnis pengelolaan sampah. Aktivitas bank sampah adalah mengumpulkan sampah yang sudah dipisahkan berdasarkan jenisnya lalu mendistribusikannya ke pabrik atau pencacah. Bank sampah sendiri adalah salah satu strategi penerapan 3R (*Reuse, Reduce, Recycle*) dalam pengelolaan sampah di tingkat masyarakat. Berbeda dengan pengepul yang fokus hanya kepada bisnis. Masyarakat yang menabung sampah di bank sampah bisa dikatakan sebagai nasabah. Nasabah bank sampah akan diberi edukasi bagaimana cara memilah sampah yang benar, sehingga bernilai jual, lalu dapat ditabungkan, itulah nilai sosial dari bank sampah. Sedangkan pengepul hanya berfokus pada pengumpulan sampah lalu dipilah sendiri oleh pegawainya lalu dijual.

Masalah yang ditemui adalah masyarakat masih memiliki paradigma kumpul, angkut, buang. Hal ini sesuai dengan undang-undang nomor 18 Tahun 2008 yang mengatur tentang pengelolaan sampah dan Peraturan Pemerintah nomor 81 Tahun 2012 yang menjelaskan bahwa perlu adanya perubahan paradigma masyarakat menjadi paradigma pengolahan sampah. Berkaitan dengan masalah masyarakat masih memiliki

paradigma kumpul, angkut, buang, masalah lainnya adalah masyarakat masih belum mengetahui apa itu bank sampah dan belum mengetahui apakah ada bank sampah disekitarnya.

Perkembangan teknologi informasi saat ini begitu cepat. Jumlah Pengguna internet saat ini cukup tinggi. Berdasarkan *survei* menurut Asosiasi Pengguna Jasa Internet Indonesia (APJII) menunjukkan bahwa pengguna internet Indonesia pada tahun 2016 memiliki 137,2 juta jiwa. Berdasarkan data tadi, masyarakat memiliki peluang untuk mengetahui dan memahami informasi tentang bank sampah. Selain masyarakat, sebenarnya dari sisi bank sampah pun akan sangat membantu dengan adanya Internet. Setelah sebelumnya mengadopsi proses bisnis secara konvensional, sudah saatnya bank sampah menggunakan aplikasi yang saling berintegrasi dengan bantuan Internet untuk bisnis bank sampah menjadi lebih efektif.

Permasalahan yang dihadapi dari sisi masyarakat atau nasabah apabila mengadopsi bisnis bank sampah secara konvensional adalah (1) masyarakat tidak mengetahui informasi bank sampah sekitar, (2) masyarakat lama untuk mengetahui informasi tentang transaksi sampah dan saldo tabungan mereka, (3) masyarakat masih ada yang belum mengetahui metode penabungan sampah yang diterapkan. Sedangkan dari sisi bank sampah adalah (1) pendataan administrasi seperti data nasabah, data sampah masih menggunakan buku, dan memungkinkan adanya redundansi data dan kesalahan mendata, (2) laporan transaksi seperti setor sampah dan pengambilan tabungan masih manual, kemungkinan salah hitung bisa terjadi, (3) perlu melakukan sosialisasi tentang bank sampah dan pemilahan sampah kepada nasabah, dan itu menggunakan dana banyak.

Pada tahun 2012 salah satunya, Irwan Purwanto, Dini Destiani, dan Partono melakukan penelitian tentang perancangan Sistem Informasi Transaksi Tabungan Bank Sampah di Garut. Hasil dari penelitian tersebut adalah aplikasi administrasi untuk membantu pegawai bank sampah dalam mengelola data transaksi tabungan dan pencarian data nasabah. Tetapi ada beberapa hal yang perlu ditambahkan, karena dari penelitian sebelumnya hanya menghasilkan aplikasi di sisi bank sampah saja, dan itu belum cukup untuk menyelesaikan masalah di sisi masyarakat atau calon nasabah. Hal yang perlu ditambahkan adalah dibuatkan aplikasi khusus untuk nasabah dan diintegrasikan dengan aplikasi di sisi bank sampah, agar memudahkan dalam pertukaran informasi dan proses administrasi dalam bisnis bank sampah lancar.

Pengembangan aplikasi yang saling terintegrasi akan mengatasi permasalahan yang dipaparkan diatas. Nasabah umumnya membutuhkan informasi yang mudah diakses, maka perlu aplikasi dengan *platform mobile*. Sedangkan untuk bank sampah karena kegiatannya adalah mendata administrasi nasabah, sampah, dan transaksi maka perlu ada aplikasi dengan perangkat yang dapat menampung data besar dan menunjang berjalannya aplikasi tersebut yaitu *website*.

Pengembangan selanjutnya adalah mengintegrasikan kedua *platform* tersebut agar saling berkomunikasi, yaitu dengan membuat *web service*. Menghubungkan dua *platform* atau lebih dengan satu pusat data (*datacenter*) akan lebih efektif dan efisien. Selain itu, apabila ingin dikembangkan lebih jauh lagi misalkan akan diintegrasikan dengan *smartcity* ataupun *Internet of Things (IoT)* hal itu akan lebih mudah dengan mengakses *web service* yang sudah dikembangkan. Tujuan utamanya adalah memudahkan proses bisnis bank sampah dan dengan teknologi dapat memudahkan masyarakat atau nasabah dalam mengakses informasi, konten edukasi dan bisnis transaksi sampah.

Web service memiliki dua metode yang berorientasi pada sumber daya informasi dan layanan informasi yaitu *Simple Object Access Protocol (SOAP)* dan *Representational State Transfer (REST)*. Sudah banyak perusahaan yang menggunakan metode SOAP, tetapi saat ini metode REST adalah yang paling dominan digunakan untuk pengembangan *web service* dikarenakan lebih efektif dalam pengembangan dan penggunaan. Format *response* yang dimiliki SOAP hanya berupa XML, berbeda dengan REST yang dapat memberikan *response* berupa XML, *JavaScript Object Notation (JSON)* ataupun format *text* lainnya [1].

2.1. Metode Pengumpulan Data

a.) Teknik Wawancara

Teknik wawancara adalah salah satu teknik pengambilan data dengan cara mendapatkan informasi secara langsung dengan mengajukan pertanyaan antara pewawancara dengan narasumber [2]. Unsur dalam teknik wawancara perlu dipilih secara baik, mulai dari narasumber, pertanyaan yang akan ditanyakan, dan pewawancara. Hal itu dikarenakan, akan sangat berpengaruh pada kualitas data yang akan didapat.

2.2. Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan adalah *Iterative Incremental*. *Iterative Incremental* adalah metode pengembangan sistem yang menggabungkan metode pengembangan *iterative* dan *incremental*. Metode *iterative* adalah kegiatan yang berfokus pada revisi secara berulang setiap pengembangan telah mencapai target. Sedangkan metode *incremental* adalah metode pengembangan sistem

yang berfokus pada proses pengembangan aplikasi yang dikerjakan dari satu tahap ke tahap selanjutnya sampai pada waktu yang telah ditentukan [3].

Pada metode *Iterative Incremental* dapat diterapkan siklus pengembangan sistem. Siklus tersebut memiliki empat fase [4], yaitu :

- a.) *Inception Phase*
 Fase ini adalah fase untuk melakukan interaksi dengan pengguna untuk menentukan identifikasi kebutuhan sistem dan fitur yang akan dibuat. Proses yang dilakukan pada fase ini adalah melakukan wawancara dan mengajukan kuisisioner kepada bank sampah dan nasabah.
- b.) *Elaboration Phase*
 Fase ini adalah proses analisis dan desain sistem. Proses desain sistem pada umumnya dirancang menggunakan UML diagram melalui *use case diagram*, *activity diagram*, *class diagram* atau *sequence diagram*.
- c.) *Construction Phase*
 Fase ini adalah fase untuk melakukan penulisan kode program *web service* sesuai dengan kebutuhan dan fitur-fitur yang sudah diidentifikasi. Setelah penulisan kode program selesai maka akan dilanjutkan proses pengujian. Setelah proses pengujian lalu akan dilakukan proses *iterative review*, proses ini dilakukan kepada bank sampah dan nasabah untuk melakukan tes fitur terhadap aplikasi yang sudah terintegrasi. Proses akan dilanjutkan kepada fase selanjutnya apabila pengguna aplikasi telah menyetujuinya.
- d.) *Transition Phase*
 Fase transisi adalah fase terakhir pada pengembangan sistem untuk mulai merilis *web service* kedalam lingkungan yang sebenarnya yaitu *server production*. Sehingga *web service* yang sudah dirilis akan bisa diakses oleh *platform mobile* dan web dan akan saling berkomunikasi.

2.3. Metode Penulisan Dokumentasi REST *Application Programming Interface* (API)

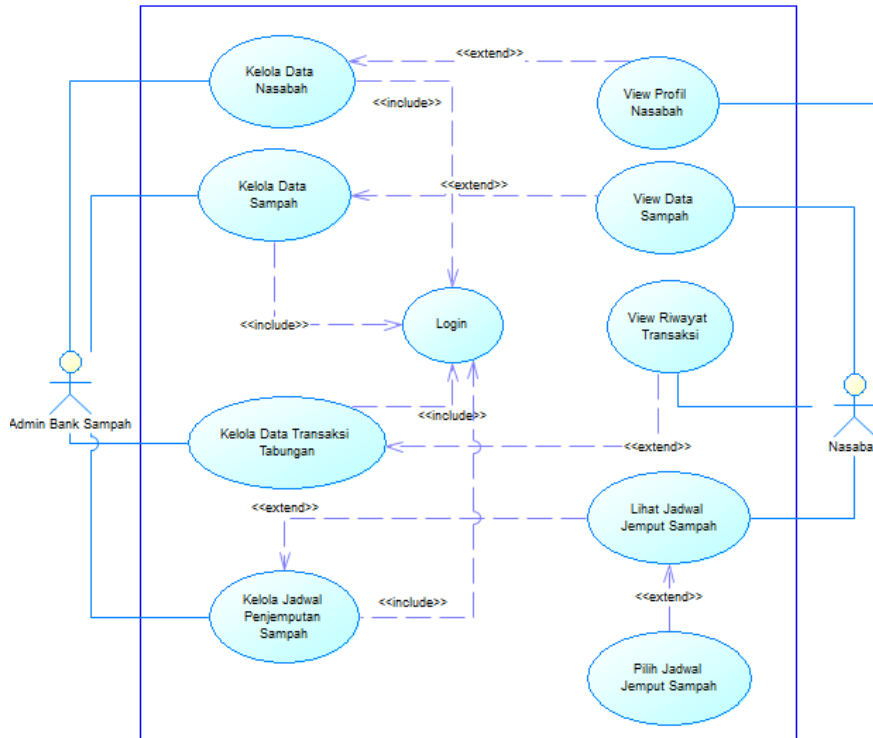
Penulisan dokumentasi untuk *web service* REST sangat penting untuk pengembangan selanjutnya. Tujuan dari pembuatan dokumentasi ini adalah meminimalkan ketidakfahaman *developer* dan agar proses konsumsi *endpoint* dari API berjalan lancar dan konsisten

Tabel 1. Struktur penulisan dokumentasi REST API

Komponen	Keterangan	Contoh
<i>Title</i>	Nama API	List Data Nasabah
<i>URL (endpoint)</i>	Struktur URL (path)	/list_nasabah atau /list_nasabah?id_bs=:id
<i>Method</i>	Tipe <i>request</i>	GET / PUT / POST / DELETE
<i>Header Params</i>	Optional, untuk memberikan identitas pada setiap transaksi pertukaran data	Authorization : Bearer <token>
<i>URL params</i>	Mendeklarasikan parameter yang akan dikirim melalui URL ketika <i>request</i> . Pisahkan dengan parameter wajib atau opsional	Required : id_bs = [integer], id_bs = 1 Optional : -
<i>Data params</i>	Mendeklarasikan parameter yang akan dikirim melalui <i>body</i> ketika <i>request</i> .	{ data : { nama : [String] } }, { data : { nama : "widyan" } }
<i>Success Response</i>	Menampilkan hasil <i>response</i> sukses apabila proses <i>request</i> berhasil	code : 200 status : true message : "data sukses ditampilkan" result : { id = 2 }
<i>Error Response</i>	Menampilkan hasil <i>response</i> gagal apabila proses <i>request</i> berhasil	code : 404 status : false message : "Unauthorized" result : null
<i>Notes</i>	Komentar atau penjelasan singkat tentang API	API untuk mendapatkan list data nasabah

3. Hasil dan Pembahasan

Hasil dari penelitian ini adalah *web service* yang mengintegrasikan aplikasi *mobile* dan *website* pada bank sampah. Berdasarkan hasil wawancara, dapat disimpulkan bahwa fitur-fitur yang terdapat pada aplikasi bank sampah adalah seperti pada Gambar 1.

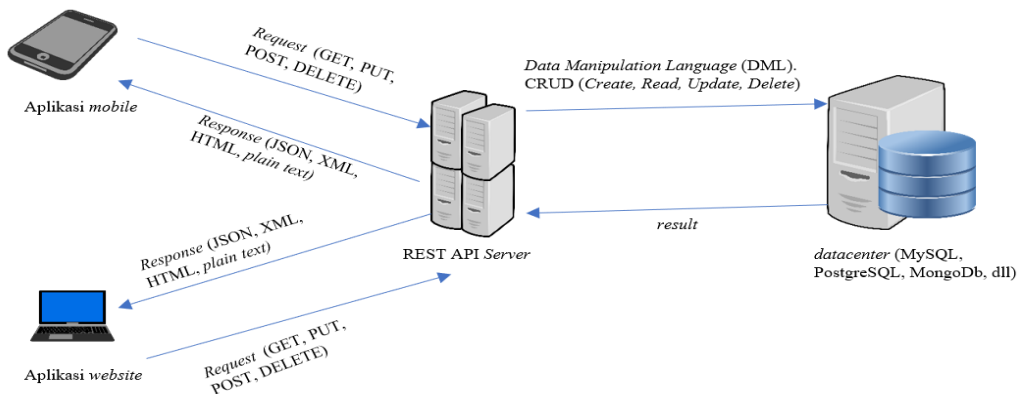


Gambar 1. Use case aplikasi pada bank sampah

Aktor yang terlibat dalam proses bisnis bank sampah adalah admin bank sampah dan nasabah. Setiap aktor memiliki *platform* aplikasi masing-masing, untuk admin bank sampah menggunakan aplikasi berbasis *website* dan *mobile* untuk proses administrasi data dan penabungan sampah. Sedangkan untuk nasabah menggunakan *platform mobile* untuk melakukan aktivitas bisnis bank sampah.

Aktivitas yang dilakukan admin bank sampah lebih kepada proses administrasi data dan proses penabungan sampah. Proses administrasi data bank sampah meliputi kelola data nasabah, kelola data sampah dan kelola data transaksi. Proses penabungan sampah meliputi kelola jadwal penjemputan sampah dan setor sampah. Sedangkan aktivitas yang dilakukan oleh nasabah adalah melihat informasi tentang data diri, data sampah, data riwayat transaksi, data jadwal jemput sampah dan memilih jadwal jemput sampah.

Struktur perancangan *web service* REST API yang akan dibuat adalah mengintegrasikan aplikasi *mobile* dengan *website* agar saling berkomunikasi dengan hanya satu pusat data (*datacenter*) saja. Struktur *web service* REST API dapat dilihat pada Gambar 2.



Gambar 2. Struktur *web service* REST API

Fitur pada aplikasi admin bank sampah umumnya adalah untuk administrasi data. Proses yang biasa dilakukan adalah input, ubah, hapus, dan lihat data. Struktur API untuk fitur seperti kelola data nasabah, kelola data sampah, kelola data transaksi tabungan dan kelola data jadwal penjemputan sampah dapat dilihat pada Tabel 2. Khusus untuk kelola data transaksi tabungan kemungkinan proses yang dilakukan adalah hanya input data dan lihat data. Karena, apabila ada proses ubah dan hapus memungkinkan akan merusak proses transaksi yang terjadi, dan mengakibatkan data tidak murni valid.

Tabel 2. Struktur API untuk proses Administrasi data dan lihat data pada aplikasi bank sampah dan nasabah

Komponen	Administrasi data	View data
<i>Method</i>	Input : POST / DELETE Ubah : POST / PUT Hapus : DELETE / POST	GET
<i>Header params</i>	Authorization : Bearer <token>	Authorization : Bearer <token>
<i>URL params</i>	Required : id_bs = [integer], id_bs = 1 Optional : -	Required : id_bs = [integer], id_bs = 1 Optional : -
<i>Data params</i>	- { data : { nama : [String] } }, { data : { nama : "widyan" } }	{ data : { nama : [String] } }, { data : { nama : "widyan" } }
<i>Success Response</i>	code : 200 status : true message : "data sukses ditampilkan" result : { id = 2 }	code : 200 status : true message : "data sukses ditampilkan" result : { data : { id : 2 nama_bs : "banksampah bersinar", alamat_bs : "Jl. Ciwastra", } }
<i>Error Response</i>	code : 404 status : false message : "Unauthorized" result : null	code : 404 status : false message : "Unauthorized" result : null

Pada proses administrasi data input, ubah dan hapus pada umumnya menggunakan API *method* POST, PUT, dan DELETE walaupun sebenarnya bisa juga menggunakan *method* GET. Tetapi apabila menggunakan *method* GET untuk proses administrasi terdapat kelemahan, yaitu parameter data yang dikirimkan tidak bisa menggunakan *data params*, melainkan harus menggunakan *url params*. Perbedaan menggunakan *url params* dan *data params* adalah jika menggunakan *url params*, nama parameter dan data yang dikirimkan akan terlihat pada URL, sedangkan jika menggunakan *data params* nama parameter dan data yang dikirimkan akan tidak terlihat pada URL. Kelemahan yang didapat apabila menggunakan *method* GET untuk proses administrasi data adalah data dapat di lihat pada URL dan itu tidak aman.

Sebaliknya untuk proses lihat data umumnya menggunakan API *method* GET. Untuk memutuskan apakah akan menggunakan *method* GET atau POST, perlu diketahui dahulu apakah akan ada perubahan data pada *server* atau tidak. Jika, akan ada perubahan data maka menggunakan POST, tetapi jika tidak akan ada perubahan data maka menggunakan GET.

Penentuan parameter yang dikirimkan tergantung pada hasil *response* yang diinginkan. Hal ini perlu disesuaikan dengan masing-masing fitur. Begitu juga untuk menentukan hasil *response*, tergantung pada parameter yang dikirimkan kepada *server*. Seperti ilustrasi pada Gambar 2, API pada *server* akan memproses parameter yang dikirimkan kepada *datacenter*, lalu *datacenter* akan memberikan *result*, selanjutnya API akan memberikan *response* yang sesuai dengan *request* yang dikirimkan.

Keamanan dan otentikasi pada proses pertukaran data pada API sangat penting. Maka dari itu perlu adanya *header params* untuk setiap *request*. Fungsi dari *header params* adalah memberikan identitas untuk setiap *request*. Umumnya *key* parameter yang dominan digunakan adalah "Content-Type" dan "Authorization". *Content-Type* menunjukkan jenis format *response* yang akan diberikan, apakah itu JSON, XML, HTML, atau *plain text*. Sedangkan untuk *Authorization* adalah untuk memastikan bahwa *request* berasal dari asal/*user* yang sama, berfungsi untuk keamanan pada API.

Komponen URL params perlu disesuaikan dengan kebutuhan.

Hasil dari penelitian ini adalah menghasilkan *end point* API yang ada di setiap modul aplikasi untuk nantinya dapat digunakan dalam mengintegrasikan ke berbagai *platform* aplikasi untuk bank sampah dan nasabah. Pada aplikasi bank sampah terdiri dari *end point* modul kelola sampah, kelola nasabah, kelola jadwal jemput sampah, dan transaksi tabungan. Sedangkan, untuk nasabah terdiri dari *end point* untuk lihat profil, lihat data sampah, lihat riwayat transaksi, lihat jadwal jemput sampah dan memilih jemput sampah.

Keuntungan yang didapat apabila mengimplementasikan *web service* untuk mengintegrasikan aplikasi *mobile* dengan *website* pada bank sampah adalah :

- i. Pertukaran informasi dan proses administrasi akan menjadi efektif antara bank sampah dan nasabah.
- ii. Mudah untuk melakukan integrasi dengan *platform* lainnya, ataupun di integrasikan dengan *smartcity* atau IoT.
- iii. Mengurangi biaya pembuatan aplikasi yang terpisah.
- iv. Biaya implementasi relatif murah, karena bersifat integrasi tidak perlu menyewa banyak *server* karena hanya ada satu *datacenter*.

3.1. Cara Pengujian Web Service

Untuk pengujian *web service* dapat menggunakan *tools* yang umum digunakan oleh para *developer* seperti Postman atau SoapUi. Metode yang digunakan untuk menguji *web service* dapat menggunakan *Black box testing*. *Black box testing* adalah pengujian dilakukan dengan mengeksekusi data uji secara fungsional pada perangkat lunak apakah bekerja dengan baik atau tidak [5]. Beberapa hal umum yang perlu diuji pada *web service*, seperti :

- i. Keamanan (*Security*) : Dari sisi keamanan, perlu diuji seputar token, *authorization* dan *method* pertukaran data.
- ii. *User Input* : Dari sisi *user input*, perlu diuji seputar data yang akan dikirim, tipe data dan *schema* format data.

Response API : Dari sisi *response* API, perlu diuji tentang hasil (*result*) pengolahan data dan harus sesuai dengan *schema* yang sudah ditentukan pada saat *request*.

4. Simpulan

Paper penelitian ini menjelaskan tentang perancangan *web service* untuk mengintegrasikan aplikasi *mobile* dan *website* yang sudah ada pada bank sampah. Metode *web service* yang digunakan adalah yang sudah dominan digunakan saat ini karena mudah untuk diimplementasikan yaitu REST API. Fitur aplikasi yang sudah ada seperti proses administrasi data bank sampah meliputi kelola data nasabah, kelola data sampah dan kelola data transaksi. Proses penabungan sampah meliputi kelola jadwal penjemputan sampah dan setor sampah. Lalu fitur untuk nasabah seperti melihat informasi tentang data diri, data sampah, data riwayat transaksi, data jadwal jemput sampah dan memilih jadwal jemput sampah. Semua proses pertukaran informasi yang ada pada setiap fitur pada *platform* yang berbeda akan diintegrasikan dengan *web service*. Hal ini akan sangat membantu untuk berjalannya bisnis bank sampah, karena akan lebih efektif dalam hal proses administrasi dan pertukaran informasi dengan nasabah. Selain itu, untuk pengembangan kedepannya akan sangat mudah jika akan di integrasikan dengan *smartcity* ataupun perangkat IoT karena mengimplementasikan *web service* REST API.

Daftar Pustaka

- [1] Sutanta, E., & Mustofa, K.s. Identifying The Needs of Web Service to Data Synchronization Between Information Systems as E-Government Ecosystem at Bantul-Yogyakarta. Bandung – Teknik Informatika STMIK Bandung; 2012
- [2] Djaelani, A. R. Teknik Pengumpulan Data Dalam Penelitian Kualitatif. Semarang- FPTK IKIP Semarang; 2013
- [3] Cockburn, A. Using Both Incremental and Iterative Development. *CrossTalk: The Journal of Defense Software Engineering*. 2008; 21(5): 27–30.
- [4] Larman, C. Applying UML and Patterns. 2nd edition. 2002
- [5] Wibisono, W., & Baskoro, F. Pengujian Perangkat Lunak Dengan Menggunakan Model Behaviour Uml. (2002); 1: 43–50.