Integrasi Aplikasi Desktop dan Mobile pada Koperasi Karyawan Mitra Makmur

Muchamad Akbar Nurul Adzan[1], De Rosal Ignatius Moses Setiadi[2], Desi Purwanti Kusumaningrum[3], Eko Hari Rachmawanto[4], Christy Atika Sari[5]
Program Studi Ilmu Komputer; Universitas Dian Nuswantoro Semarang
Jl. Nakula I No 5-11, Pendrikan Kidul, Semarang Tengah, Kota Semarang. Telp. 024-3517261
e-mail: [1]111201307488@mhs.dinus.ac.id, [2]moses@dsn.dinus.ac.id, [3]desi.purwanti@dsn.dinus.ac.id, [4]eko.hari@dsn.dinus.ac.id, [5]atika.sari@dsn.dinus.ac.id

Abstrak

Koperasi Karyawan Mitra Makmur memiliki aplikasi Dekstop untuk menunjang kegiatannya. Walaupun demikian terdapat kendala dimana kurang praktisnya aplikasi berbasis desktop karena hanya dapat dibuka melalui komputer saja. Saat ini banyak anggota koperasi yang menggunakan perangkat mobile, hal ini melatarbelakangi perlunya pembuatan aplikasi berbasis mobile. Maka dibutuhkan aplikasi mobile dibutuhkan sebagai solusi untuk memudahkan akses. Karena terdapat dua macam aplikasi, maka keduannya harus diintegrasikan. Salah satu cara untuk mengintegrasikan kedua aplikasi tersebut dengan web service dan backend sebagai sistem manajemen konten pada aplikasi mobile. JSON (Javascript Object Notation) diusulkan sebagai perangkat pertukaran data. JSON dipilih karena prosesnya yang cepat dan ringan. Untuk mengukur kinerja web service pada proses integrasi, maka perlu dianalisis dan diukur kebutuhan waktu request HTTP sampai didapatkan response HTTP. Terdapat dua cara pengukuran lama waktu request web service yaitu dengan pengujian secara online dan lokal. Masing-masing pengujian request dilakukan sebanyak lima kali dimana didapat rata-rata 1192 ms. Waktu yang dibutuhkan tersebut relatif wajar dan dapat disimpulkan bahwa proses integrasi dapat berjalan dengan baik.

Kata kunci: *Web Service, Javascript Object Notation, Waterfall, Backend.*

Abstract

Koperasi Karyawan Mitra Makmur has a desktop application to support its activities. However, there are obstacles where the less practical desktop-based applications because it can only be opened through a computer only. Currently, many members of cooperatives who use mobile devices, this is behind the necessity of making a mobile-based application. So required mobile application needed as a solution to easy access. Because there are two kinds of applications, the two must be integrated. One way to integrated both applications with web service and backend as a content management system in mobile applications. JSON (Javascript Object Notation) is proposed as a data exchange device. JSON was chosen because the process is fast and light. To measure the performance of web services on the integration process, it needs to be analyzed and measured the need of HTTP request time to get response HTTP. There are two ways of measuring the duration of a web service request is by testing online and local. Each request test is done five times, which is an average of 1192 ms. The time required is relatively reasonable and can be concluded that the integrity process can run well.

Keywords*: Web Service, Javascript Object Notation, Waterfall, Backend.*

1. Pendahuluan

Koperasi Karyawan Mitra Makmur saat ini telah memiliki aplikasi *desktop* yang berguna sebagai penunjang dalam kegiatan yang berkaitan dengan koperasi. Tetapi aplikasi ini tidak dapat dimanfaatkan secara maksimal mengingat ketika menggunakan aplikasi desktop dibutuhkan komputer atau laptop untuk membukanya sehingga pengecekan tidak dapat dilakukan setiap saat. Dalam upaya optimalisasi penggunaan aplikasi khususnya dalam menyajikan informasi yang lebih *up to date* bagi karyawan atau peserta koperasi yang memiliki mobilitas tinggi, maka dibutuhkan pengembangan aplikasi *mobile*. Aplikasi *mobile* tentunya tidak serta merta menghilangkan fungsi aplikasi desktop, sehingga aplikasi desktop yang telah ada tetap dapat digunakan khususnya bagi *admin* koperasi. Karena nantinya akan terdapat dua basis aplikasi, maka kedua aplikasi perlu diintegrasikan. Untuk mengintegrasikan kedua aplikasi tersebut dibuat

 DOI: 10.30864/eksplora.v8i1.142

web service dan *backend* berbasis *web* sebagai manajemen konten dan data yang ada pada aplikasi *mobile*[1]. Sehingga dapat terbentuk sistem yang saling terhubung yaitu antara aplikasi *desktop*, *web* dan *mobile*.

Web service adalah serangkaian komponen peranti lunak yang bertukar informasi antara satu sama lain dengan bebas menggunakan standar komunikasi web dan bahasa standar [2]. *Web service* dapat bertukar informasi pada dua sistem yang berbeda, terlepas dari sistem operasi dan bahasa yang digunakan. *Web service* dirancang untuk mendukung interaksi yang bisa beroperasi *machine-to-machine* diatas jaringan. *Backend* adalah halaman *dashboard admin* atau sering disebut dengan CMS (*Content Management System*) yang berguna sebagai tempat perubahan informasi atau konten pada suatu aplikasi dengan cara masuk sebagai administrator [3].

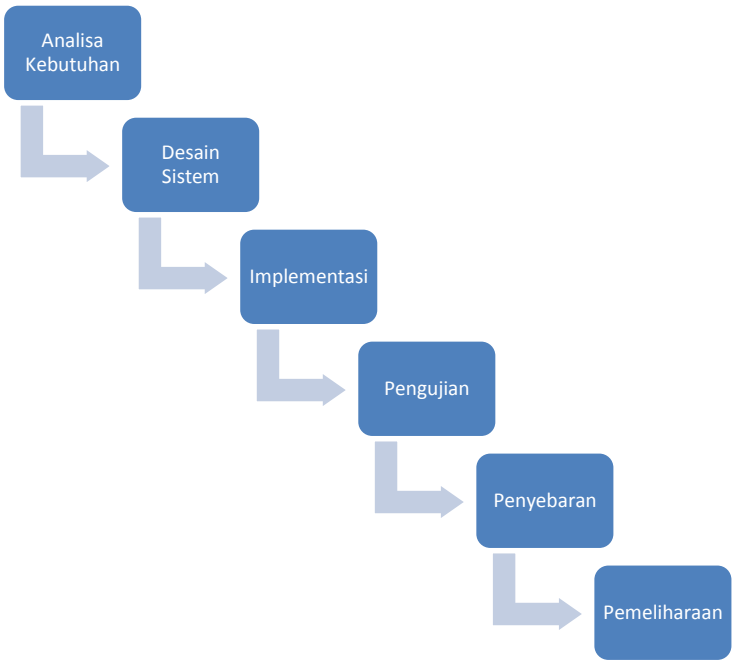
Pada penelitian yang dilakukan oleh Martinus Raditia Sigit Surendra tahun 2014 yang berjudul "*Implementasi PHP Web Service Sebagai Penyedia Data Aplikasi Mobile*" [4]. Hasil penelitian dengan pemanggilan *method* menggunakan XML (*Extensible Markup Language*) lebih rumit dan memiliki ukuran *file* besar dibandingkan dengan HTTP (*Hypertext Transfer Protocol*) *request* dengan penambahan layer PHP yang diimplementasikan dalam aplikasi *mobile* dan dengan menggunakan format data JSON (*Javascript Object Notation*) dinilai memiliki ukuran *file* kecil dan *response* yang diterima lebih cepat.

Pada penelitian yang dilakukan oleh Ilhamsyah tahun 2011 yang berjudul "*Implementasi Web Service Sistem Integrasi Data Menggunakan Teknik Replikasi Data Pada Inventarisasi Bangunan Pemerintah*" [5]. Hasil dari penelitian di sini adalah proses integrasi menggunakan *web service* pada BPKAD kota atau kabupaten dengan melalui *service provider* yang diakses oleh *client* dan kemudian dilakukan proses replikasi data ke BPKAD provinsi Kalimantan Barat. Teknologi *web service* dengan menggunakan teknik replikasi data sangat efektif pada transaksi data, karena *method* dari *web service* akan secara otomatis memproses data yang dikirim ke *server*.

Berdasarkan beberapa penelitian terkait di atas maka penelitian ini mengintegrasikan aplikasi *Dekstop* dan Mobile pada Koperasi Karyawan Mitra Makmur. Data yang digunakan merupakan data langsung dari sumber atau pihak koperasi yaitu berupa data peserta, transaksi, tabungan, info potongan, pinjaman, simpanan dan (jaminan kesehatan) yang nantinya data akan dikirim dalam bentuk *JSON* dengan *web service* dari aplikasi *desktop* milik pihak Koperasi Karyawan Mitra Makmur.

2. Metode Penelitian

Dalam penelitian ini model atau metode yang digunakan adalah *waterfall model*. Pada *waterfall* memiliki ciri khas dimana setiap fase atau tahapan akan diselesaikan satu persatu sebelum berlanjut ke tahap atau fase berikutnya dan hasil dari tahap sebelumnya akan digunakan sebagai jalan masuk ke fase atau tahap selanjutnya [6]. Adapun proses tahapan dari *waterfall* pada gambar berikut.



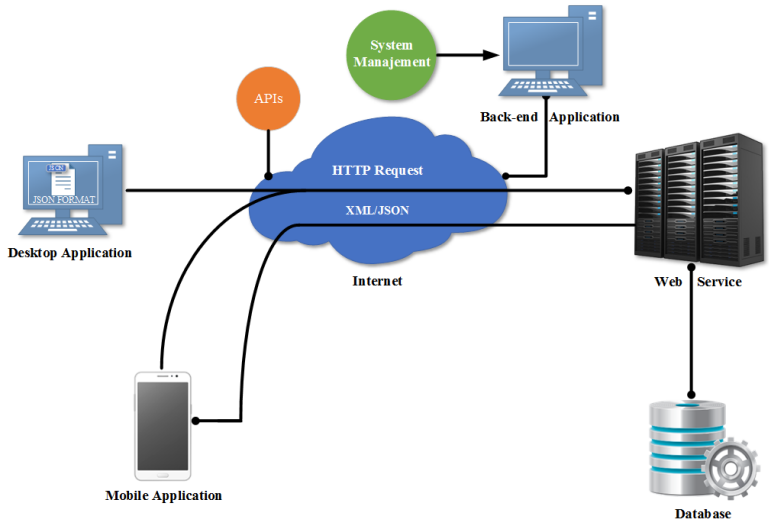
Gambar 1. Struktur model waterfall [3].

Proses model dari Gambar 1 yaitu adalah tahap-tahap yang ada pada model *waterfall* yang akan diimplementasikan dan diuji sesuai dengan spesifikasi dari sistem *web service* dan *backend* yang dibuat. Sebagai berikut proses pada masing-masing tahapan.

2.1. Analisa Kebutuhan

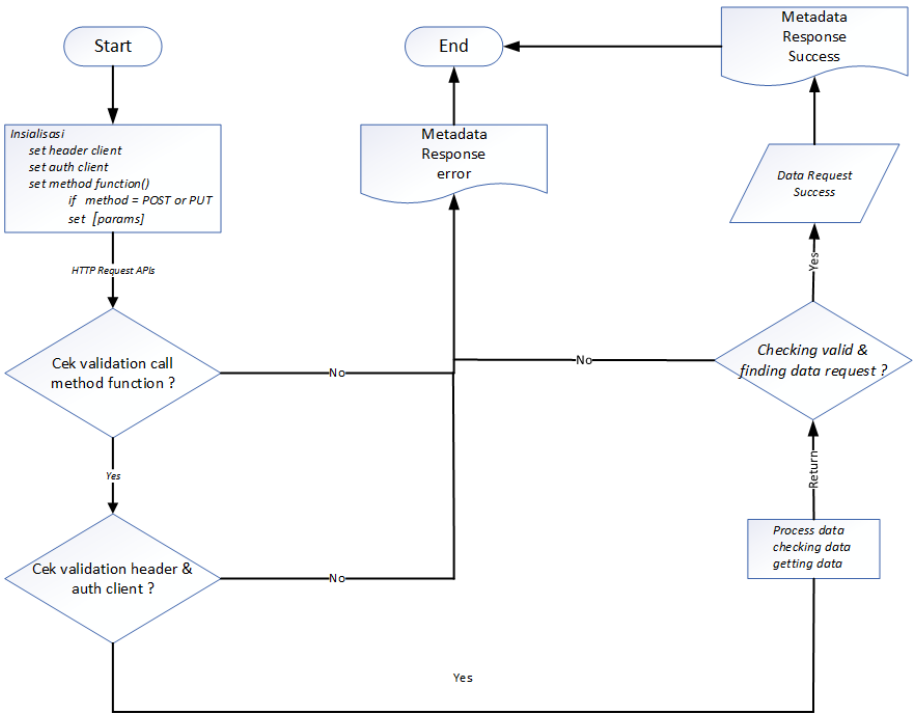
Pada tahap ini semua persyaratan atau analisa kebutuhan akan didokumentasikan. Seluruh kebutuhan dalam membuat produk harus disajikan dalam tahap ini. Kebutuhan informasi yang biasanya didapat melalui *interview*, *survey* atau diskusi.

2.2. Desain Sistem

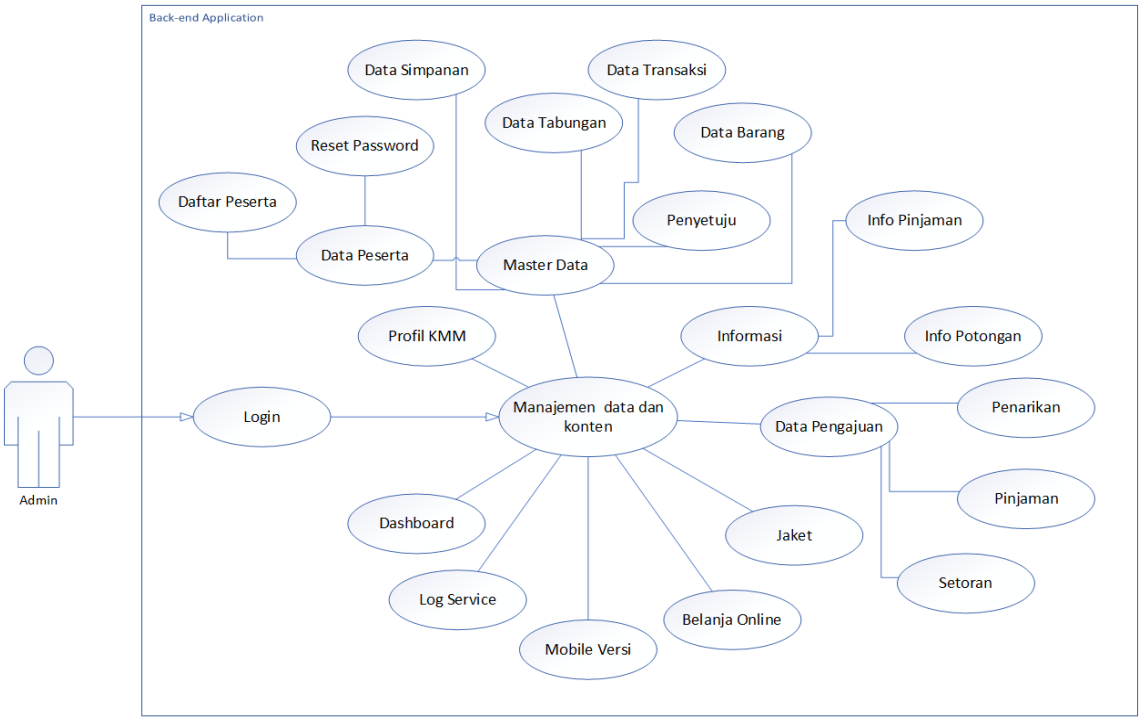


Gambar 2. Desain integrasi sistem.

Pada tahap sistem desain akan sangat membantu dalam pembuatan struktur program. Pada tahap ini menjelaskan gambaran atau alur sistem keseluruhan apa saja yang akan dikerjakan dan seperti apa tampilannya. Gambar 3 dan Gambar 4 merupakan desain sistem yang digunakan dalam pembuatan aplikasi *backend* dan *web service*.



Gambar 3. Desain *flowchart web service*.



Gambar 4. *Use case* diagram *backend*.

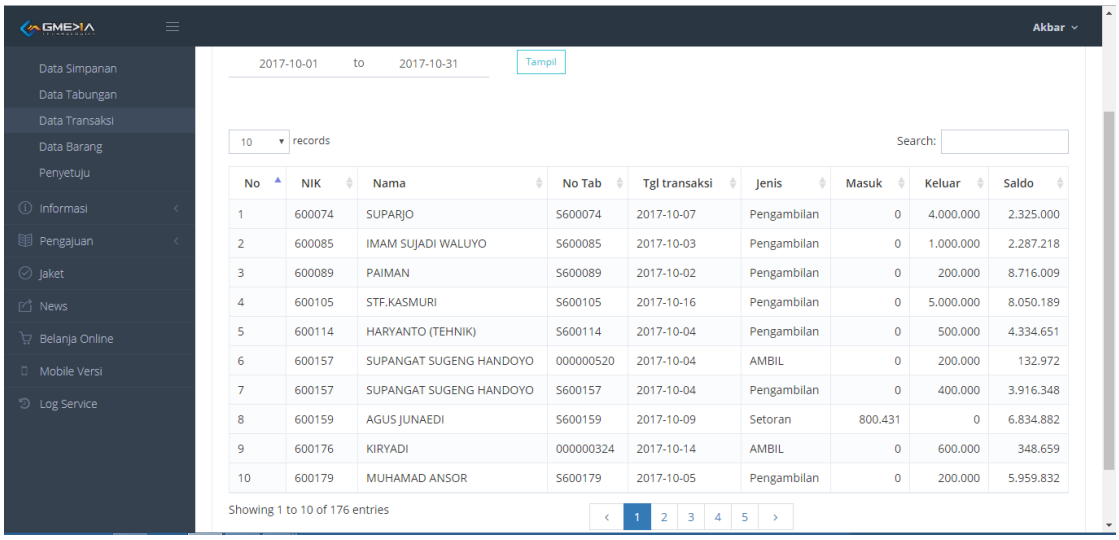
2.3. Implementasi

Pada tahap implementasi desain *interface* akan diimplementasikan pada aplikasi dengan *operating system Android*, sedangkan aplikasi *backend* dan *web service* akan dibuat berbasis *website* menggunakan JSON.

2.4. Pengujian

Tahap pengujian adalah tahapan dimana setiap unit program atau aplikasi yang telah diimplementasikan diuji. Pengujian yang dilakukan pada web *service* yaitu menggunakan aplikasi bantuan sebagai GUI *API Caller* yaitu *postman* dan pada aplikasi *backend* dilakukan penambahan konten pada aplikasi *mobile*. Pada tahap ini akan dijelaskan secara lebih detail pada bagian hasil dan pembahasan.

2.5. Penyebaran



Gambar 5. Data transaksi bulan Oktober 2017.

Pada tahap ini merupakan tahapan dimana aplikasi yang sudah diuji pada keseluruhan sistem, maka dilanjut dengan melakukan penyebaran produk pada lingkungan *user* Koperasi Karyawan Mitra Makmur itu sendiri. Aplikasi ini sendiri sudah di *hosting* pada *server* milik Gmedia Semarang sebagai rekan kerja dari pihak koperasi, dan aplikasi ini sudah digunakan dan berjalan sampai sekarang. Gambar 5 merupakan contoh data transaksi yang diambil dari aplikasi *backend*.

2.6. Pemeliharaan

Pada tahap pemeliharaan kondisi sistem akan terus di cek dan di-*maintenance* sesuai dengan kebutuhan untuk meminimalisir masalah yang ada saat sistem diimplementasikan.

3. Hasil dan Pembahasan

3.1. Implementasi

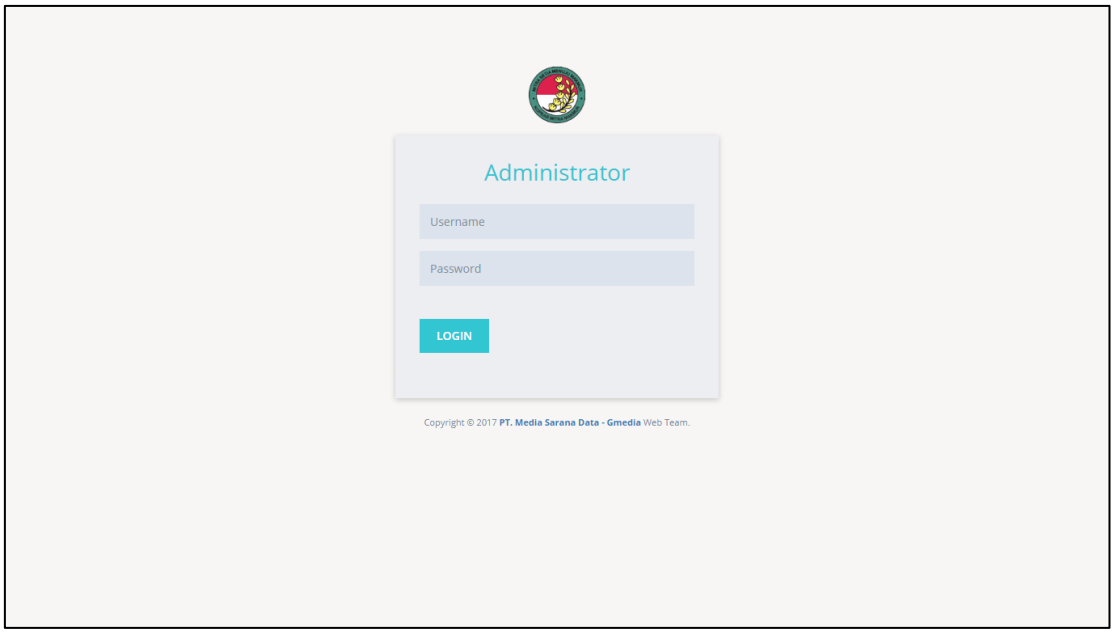
Implementasi atau penerapan kinerja aplikasi dan contoh kode (*coding*) pada *web service* dan desain *interface* yang ada pada aplikasi *backend*. Sebagai pada Gambar 6 dan 7 merupakan contoh pembuatan fungsi web servis dengan *method* GET dan *method* POST, gambar 8 merupakan contoh hasil implementasi desain *interface*, sedangkan pada Gambar 9 merupakan contoh implementasi *web service*.

```
function get_tabungan($nik='')
{
    $method = $_SERVER['REQUEST_METHOD'];
    if ($method != 'GET')
    {
        $this->MyModel->bad_request();
    } else
    {
        $check_auth_client = $this->MyModel->check_auth_client();
        if ($check_auth_client == true)
        {
            $data = $this->Model_data->tampil_tabungan($nik);

            if(count($data)>0)
            {
                $json = $this->MyModel->success($data);
            }
            else
            {
                $json = $this->MyModel->data_not_found();
            }

            json_output(200, $json);
        }
    }
}
```

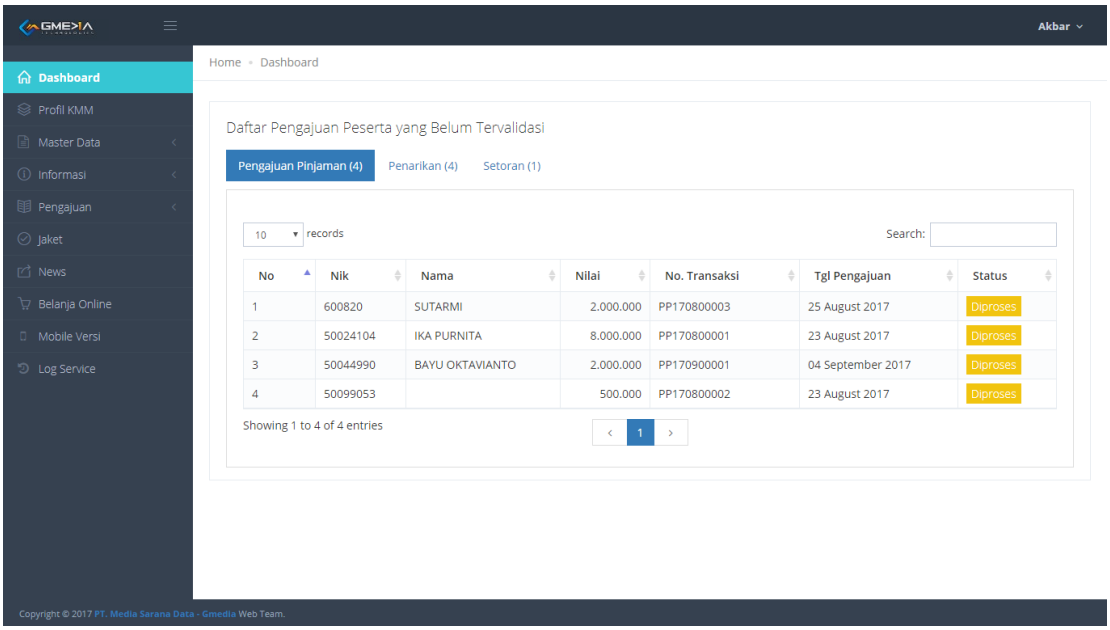
Gambar 6. Fungsi *web service method get*.

```
function get_transaksi_nik()
{
    $method = $_SERVER['REQUEST_METHOD'];
    if ($method != 'POST')
    {
        $this->MyModel->bad_request();
    } else
    {
        $check_auth_client = $this->MyModel->check_auth_client();
        if ($check_auth_client == true)
        {
            $params = json_decode(file_get_contents('php://input'), TRUE);
            $nik = $params['nik'];
            $no_tab = $params['no_tabungan'];

            $data = $this->Model_data->get_detail($nik, $no_tab);

            if(count($data)>0)
            {
                $json = $this->MyModel->success($data);
            } else
            {
                $json = $this->MyModel->data_not_found();
            }

            json_output(200, $json);
        }
    }
}
```

Gambar 7. Fungsi *web service method post*.



Gambar 8. Halaman *login backend*.



Gambar 9. Halaman *dashboard backend*.

3.2. Analisis Kinerja Web Service

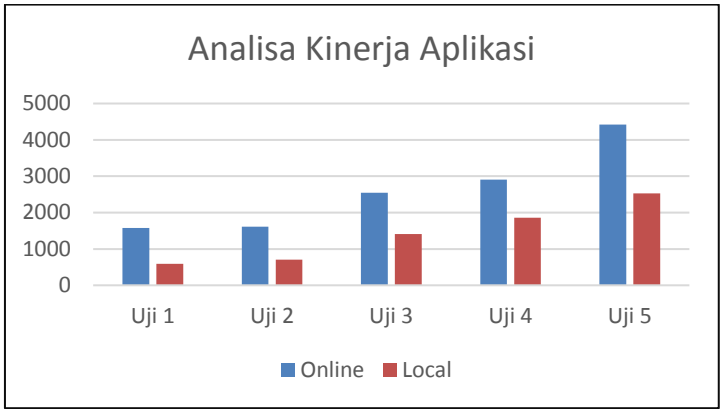
Pada bagian analisis kinerja *web service* dilakukan guna untuk mengetahui lama waktu yang dibutuhkan dalam melakukan *HTTP Request* sampai mendapatkan *HTTP Response*. Pada analisis ini dilakukan dengan cara mengambil data dalam jumlah *record* yang besar dan dilakukan dengan dua tipe yaitu dengan cara *local* dan *online*. Dalam melakukan analisis *web service* di sini menggunakan aplikasi bantuan yaitu *postman* sebagai *GUI tester web service*. Tabel 1 merupakan daftar hasil beberapa pengujian *web service* secara *local* dan *online*.

Tabel 1. Daftar hasil pengujian *web service*.

Pengujian	Record	Online (ms)	Local (ms)	Selisih
Pengujian Pertama	1000	1576	594	982
Pengujian Kedua	3000	1612	707	905

Pengujian Ketiga	5000	2542	1413	1129
Pengujian Keempat	7000	2910	1855	1055
Pengujian Kelima	9000	4422	2531	1891

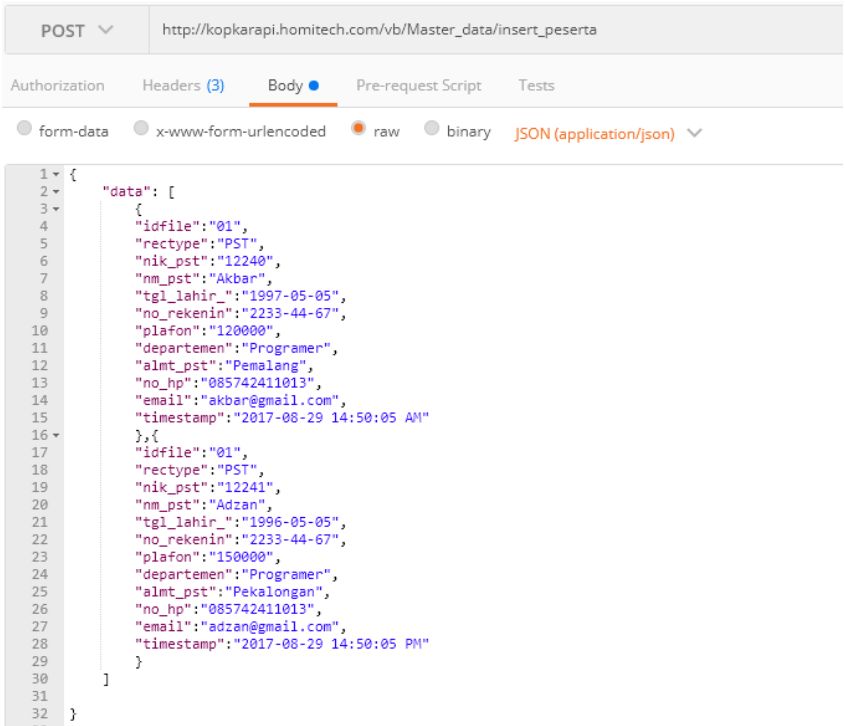
Dari hasil pengujian di atas dapat disimpulkan bahwa untuk mengukur lama waktu *response* yang dibutuhkan didapatkan selisih waktu rata-rata ketika melakukan *request* dengan cara *local* dan *online* yaitu 1192 ms (*milisecond*). Pada hasil pengujian menjelaskan bahwa waktu *request* yang lama yaitu pada bagian *online*, dikarenakan penggunaan *request* dengan *online* tergantung pada kecepatan internet yang digunakan. Adapun grafik yang didapatkan dari pengujian ditampilkan pada Gambar 10.



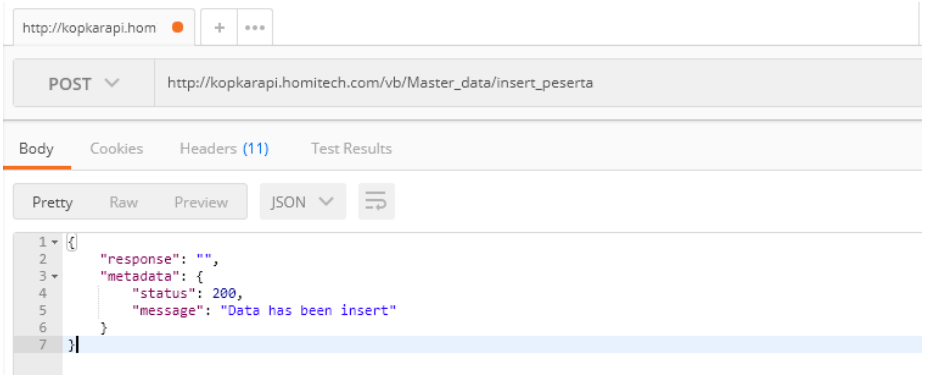
Gambar 10. Grafik hasil analisa kinerja *web service*.

3.3. Pengujian

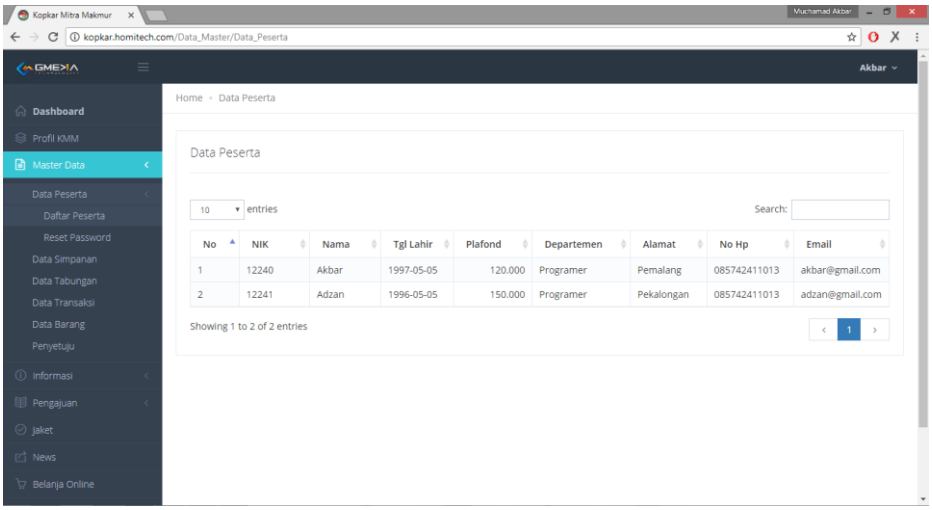
Pada bagian ini aplikasi yang telah diimplementasikan dilakukan beberapa pengujian yang dibutuhkan. Pengujian yang dilakukan di sini berupa pengujian *blackbox* pada aplikasi *backend* dan *web service*. Untuk pengujian *web service* menggunakan aplikasi *postman* dengan melakukan pengiriman data dan akan ditampilkan pada aplikasi *backend* dan untuk pengujian aplikasi *backend* dilakukan dengan mengganti atau menambahkan konten dari aplikasi *backend* yang membuat perubahan data atau konten pada aplikasi *mobile*. Berikut adapun gambar pengujian yang dilakukan.



Gambar 11. Pengaturan *value* pada *body postman*.

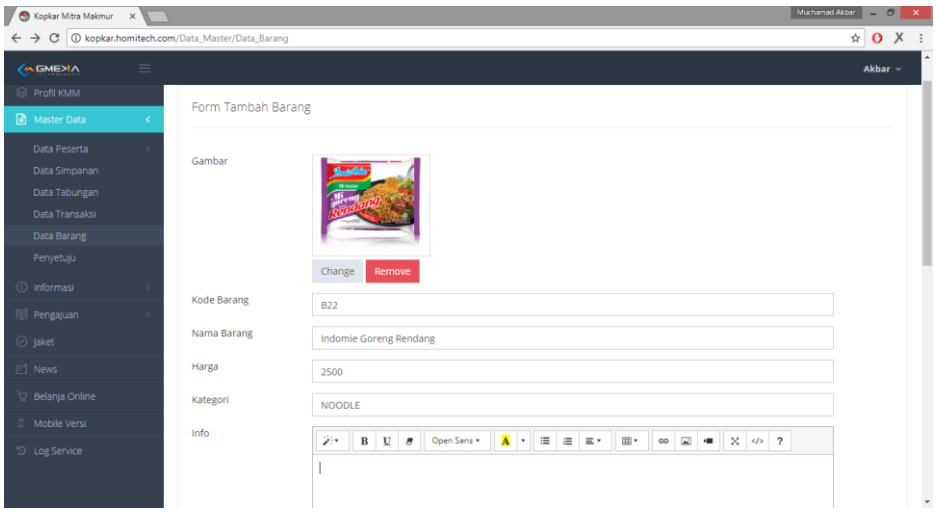


Gambar 12. Kondisi HTTP *response*.

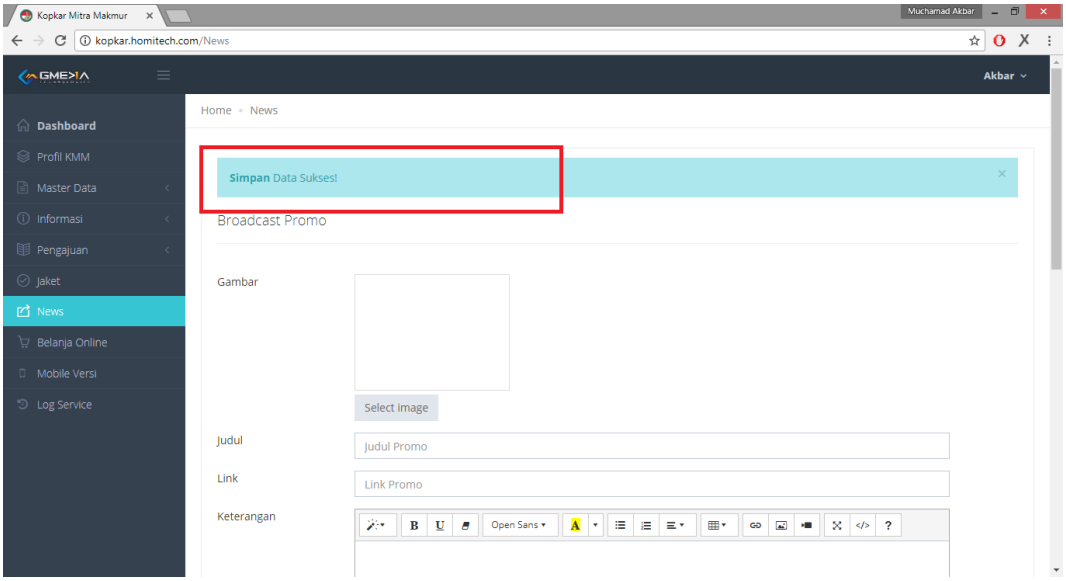


Gambar 13. Kondisi data tersimpan.

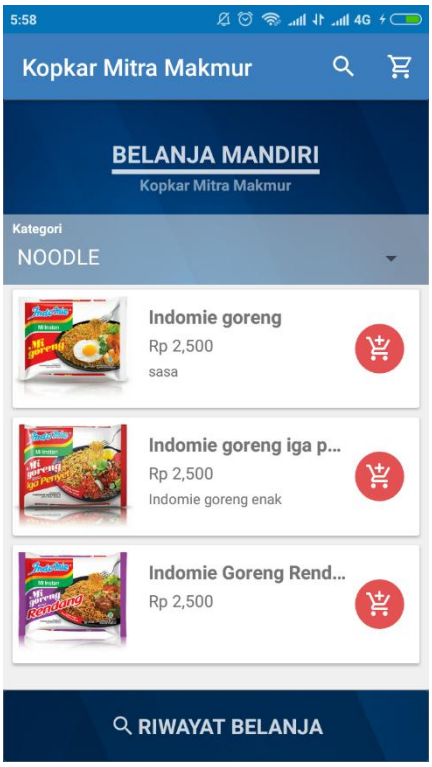
Pada pengujian *web service* yaitu dilakukan pengaturan awal pada *postman* dengan mengisi *value* yang akan dikirim pada *body* aplikasi *postman* terdapat pada Gambar 11 yaitu persiapan pengujian dengan mengisi *value* yang dikirim dan pada Gambar 12 ada kondisi *response* yang didapat pada saat selesai melalukan *request, response* yang ditujukan berupa *response* sukses yang menandakan pengiriman berhasil dan untuk hasil dari data yang dikirim melalui *postman* ada pada Gambar 13 merupakan data yang tersimpan dan ada pada aplikasi *backend*.



Gambar 14. Kondisi persiapan *backend*.



Gambar 15. Tampilan notifikasi sukses.



Gambar 16. Kondisi akhir pada aplikasi *mobile*.

Pada pengujian yang dilakukan pada aplikasi *backend* di sini adalah dengan melakukan penambahan data pada data barang, untuk persiapan penambahan data pada gambar 14 persiapan penambahan data barang dan kemudian simpan dan terdapat notifikasi sukses tersimpan pada gambar 15 yang menandakan gambar telah tersimpan dan untuk hasil yang berubah pada aplikasi *mobile* yaitu pada gambar 16. Jadi pengujian pada aplikasi *backend* merupakan menambah konten yang ada pada aplikasi *mobile*.

Perlu diketahui bahwa, aplikasi ini telah diimplementasikan dan memiliki masalah pada bagian *web service* apabila menerima *file* yang dikirim dari aplikasi *desktop* dalam satu pengiriman membawa jumlah *record* yang besar sehingga memberatkan kinerja *server* dikarenakan fungsi *web service* yang

digunakan aplikasi *desktop* memiliki validasi data per *record* dalam setiap pengiriman data. Akan tetapi dalam kendala yang dialami ini terdapat solusi dalam mengatasi masalah yang ada yaitu dengan memanfaatkan fitur dari *cpanel* yaitu fitur *cronjob* dan melakukan penambahan tabel *temporary* yang dibutuhkan. *Cronjobs* adalah sistem penjadwalan otomatis yang dibuat menggunakan *command* program yang disebut *crontab* [7].

Jadi cara kerjanya yaitu *cronjob* akan mengeksekusi fungsi yang dibuat sebelumnya secara otomatis sesuai dengan pemasangan pada sistem penjadwalan *cronjob*. *Temporary* tabel digunakan untuk media penggantian tabel utama dan dengan fitur *cronjob* yang mengeksekusi fungsi secara otomatis untuk memindah data dari tabel *temporary* menuju tabel utama.

4. Kesimpulan

Pada penelitian disini dapat disimpulkan yaitu dalam pembuatan dan perancangan pada aplikasi *backend* dan *web service* pada Koperasi Karyawan Mitra Makmur dan kemudian dilanjut dengan tahapan pengujian yang dilakukan pada *web service* yaitu terdapat pengujian pengiriman data dan analisa kinerja lama waktu yang dibutuhkan pada saat melakukan *request* dengan cara *local* dan *online* didapatkan hasil rata-rata yaitu 1190 ms (*milisecond*). Dan pada pengujian aplikasi *backend* penambahan atau perubahan data pada aplikasi *mobile* telah berjalan dengan baik dan setelah melakukan pengujian yang dilakukan, penggunaan aplikasi *backend* dan *web service* sampai sekarang masih digunakan dan berjalan dengan lancar.

Daftar Pustaka

[1] PT. Media Sarana Data, "Info tentang Perusahaan" *http://www.gmedia.net.id/pages/view/29/Info-Perusahaan*, 2012.
[2] G. Alonso, F. Casati, H. Kuno and V. Machiraju, "Web Service," in *Data-Centric Systems and Applications* , Springer, 2004, pp. 123-149.
[3] Government of the HKSAR., "Web Content Management System," *Int. J. Innov. Res. Adv. Eng.*, vol. 3, no. 3, pp. 51–56, 2016.
[4] M. Raditia and S. Surendra, "Implementasi PHP Web Service Sebagai Penyedia Data Aplikasi Mobile," vol. VI, no. 2, pp. 85–93, 2014.
[5] Ilhamsyah, "Implementasi Web Services Sistem Integrasi Data Menggunakan Teknik Replikasi Data," vol. 1, no. 2, pp. 95–104, 2011.
[6] R. S. Pressman, Software engineering: a practitioner's approach, 2014.
[7] M. O. Sandonís, "Keeper: A tool for management and automated deployment of CMS web services," in *IEEE Nuclear Science Symposium and Medical Imaging Conference (2013 NSS/MIC)*, Seoul, 2013.