



## Aplikasi Pengawasan Kendaraan di Daerah Perbatasan Entikong berbasis Teknologi PWA Dan *Load Balancing Server*

Ikhwan Ruslianto<sup>1</sup>, Uray Ristian<sup>2</sup><sup>1,2</sup>Rekayasa Sistem Komputer, Fakultas MIPA, Universitas Tanjungpura,  
<sup>1</sup>ikhwanruslianto@siskom.untan.ac.id, <sup>2</sup>eristian@siskom.untan.ac.id

### Abstract

Entikong is one of Indonesia's border areas that is crossed by vehicles across national borders from Malaysia and Brunei. Data collection is based on computers to check passing vehicles. This is of course a problem if you want to get vehicle data quickly and effectively. This will cause problems again if there is interference on the internet or the internet condition becomes slow. Therefore, it is necessary to make a mechanism for accessing application pages with offline conditions. The application of Progressive Web Apps (PWA) in an application can keep the website accessible to users even though the internet condition is offline. Service Workers on PWA will work based on an internet connection. If the application is online, data will be called from the server to use. If offline, the data will be retrieved from the device cache sent to the application. From the test results, even though the internet is offline, the Sipintar application website can still be accessed. The browser storage required is only 630kb for cache storage, 2kb for Service Worker and 302b for Indexed DB. In addition, server load balancing is needed for the use of heavy network traffic. From the results of monitoring the server shows that the usage of bandwidth resources on the Sipintar application server only has a most usage of 1.3 Mbps.

Keywords: PWA, Service Worker, Load Balancing

### Abstrak

Entikong merupakan salah satu daerah perbatasan Indonesia yang dilalui kendaraan-kendaraan lintas batas negara dari Malaysia maupun Brunei. Pendataan dilakukan berbasis komputer untuk pengawasan kendaraan yang melintas. Hal ini tentu saja menjadi masalah apabila ingin mendapatkan data kendaraan dengan cepat dan efektif. Hal ini akan menimbulkan masalah lagi apabila terjadi gangguan pada internet atau kondisi internet menjadi lambat. Oleh karena itu perlu dibuat suatu mekanisme pengaksesan halaman aplikasi dengan kondisi *offline*. Penerapan *Progressive Web Apps (PWA)* pada suatu aplikasi dapat membuat *website* tetap dapat diakses oleh pengguna meskipun kondisi internet dalam kondisi *offline*. *Service Worker* pada PWA akan bekerja berdasarkan koneksi internet. Jika aplikasi dalam kondisi *online*, maka data akan dipanggil dari server untuk digunakan. Jika *offline*, maka data akan diambil dari *cache* perangkat yang dikirimkan ke aplikasi. Dari hasil pengujian, walaupun internet dalam kondisi *offline*, *website* aplikasi Sipintar tetap dapat diakses. Adapun *storage browser* yang dibutuhkan hanya sebesar 630kb untuk *cache storage*, 2kb untuk *Service Worker* dan 302b untuk Indexed DB. Selain itu, *load balancing server* dibutuhkan untuk pemakaian *traffic* jaringan yang padat. Dari hasil pantauan *server* menunjukkan pemakaian sumber daya bandwidth pada *server* aplikasi Sipintar hanya maksimal pemakaian 1,3 Mbps.

Kata Kunci: PWA, Service Worker, Load Balancing

© 2019 Jurnal RESTI

### 1. Pendahuluan

Pengawasan kepadatan lalu lintas kendaraan antar negara sangat penting, bahkan telah menjadi prioritas sebuah lembaga pemerintah. Pemantauan lalu lintas

diperlukan untuk mengetahui kondisi kendaraan dan pengemudi secara real time dan akurat. Daerah perbatasan merupakan gerbang keluar masuknya kendaraan yang berasal dari negara lain selain

Indonesia. Setiap kendaraan yang melintas ke Indonesia, memiliki batas waktu tertentu dan harus memenuhi syarat-syarat untuk masuk ke Indonesia. Pada beberapa daerah perbatasan telah menggunakan suatu aplikasi untuk mencatat dan melakukan pengawasan kendaraan lintas batas negara. Aplikasi tersebut diakses menggunakan internet, sehingga sangat bergantung dengan kondisi internet yang stabil.

Aplikasi akan berjalan sangat lancar dan tanpa masalah saat kondisi jaringan sangat baik, tetapi gangguan terjadi biasanya ketika jaringan bermasalah. tanpa adanya notifikasi langsung ke halaman menu ketika kita menutup aplikasi web dan *loading* yang terkadang sangat lama pada jaringan yang buruk yang hanya menampilkan layar putih. Seringkali pengguna harus menginstal aplikasi native tetapi terlalu banyak aplikasi yang diinstal tentu akan semakin mengurangi ruang penyimpanan dan juga harus mencarinya terlebih dahulu di *apps store* atau *google store* sebelum dilakukan proses instalasi.

Untuk mengatasi hal tersebut, pada penelitian ini bertujuan ingin melakukan pengamatan terhadap penerapan implementasi *Progressive Web Apps* (PWA) dan *Load balancing Server*. Aplikasi PWA merupakan konsep pengembangan aplikasi berbasis web yang mencakup penerapan teknologi terbaru dari *browser* seperti *Service Workers* dan *app manifest* [1]. Konsep PWA dapat memberikan pelayanan terbaik dalam menggunakan suatu aplikasi web walaupun dalam koneksi internet yang minim atau *offline* sekalipun dengan menggunakan *Service Worker*. *Service Worker* sebagai pengatur *request* dan *response* pada sebuah aplikasi web dan dapat dirancang sesuai kebutuhan [2]. *Service Worker* digunakan strategi *caching network first*, *cache fallback* dengan tambahan *cache only* pada berkas-berkas statis. Strategi ini dipilih untuk menghindari *error token* pada kerangka kerja Laravel dan tetap mendapat respons yang cepat ketika sebuah halaman web dibuka.

Dengan konsep *offline first* dan *Web APIs* baru, aplikasi yang berjalan pada *browser* tidak akan memunculkan pesan *network error* atau *white screen* seperti yang selama ini seringkali kita alami ketika berada pada jaringan yang sangat buruk [3]. Pada PWA, ketika kita meninggalkan *browser*, dan kembali lagi membuka aplikasi tersebut pada jaringan buruk atau tanpa jaringan sekalipun kita tidak akan kehilangan UI dan data-data terakhir. *Load balancing* adalah teknik untuk mendistribusikan beban trafik pada dua atau lebih jalur koneksi secara seimbang, agar trafik dapat berjalan optimal, memaksimalkan *throughput*, memperkecil waktu tanggap dan menghindari *overload* pada salah satu jalur koneksi [4]. Metode ini diterapkan untuk mengatasi beban pemanggilan *cache* saat PWA bekerja. Dengan adanya *load balancing*, dapat mengatasi kekurangan dari metode PWA.

Penelitian terkait penerapan PWA untuk sistem aplikasi web monitoring menghasilkan respon aplikasi yang sangat baik [1]. Hasil dari penelitian ini adalah sistem aplikasi yang disisipkan *service worker* PWA mampu melakukan proses *caching* data hingga 500 data keluhan. Meskipun eksekusi waktu yang dibutuhkan dalam mengakses aplikasi lebih lama karena pemasangan *service worker*, namun aplikasi yang diakses lebih cepat ketika dalam keadaan *offline* karena data dimuat dalam *cache service worker*. Selanjutnya penelitian terkait perbandingan respon PWA dengan *Mobile Web* menghasilkan respon yang cepat dalam mengakses aplikasi web [3]. Kecepatan dalam memuat data menyesuaikan dengan ukuran berkas dan *cache* yang digunakan serta frekuensi mengakses halaman aplikasi. Pada ukuran berkas dan *cache* yang kecil *Mobile Web* masih lebih unggul dibandingkan dengan PWA, sedangkan pada ukuran berkas dan *cache* yang cukup besar PWA mampu mengungguli *Mobile Web*. Hal ini dikarenakan penyimpanan pada *Mobile Web* tidak menggunakan media penyimpanan sama sekali, sedangkan pada PWA penggunaan media penyimpanan menyesuaikan dengan *cache* yang disimpan pada peramban. Oleh karena itu keunggulan dari PWA adalah mampu bekerja dengan optimal saat menggunakan *cache* yang cukup besar dan *offline*.

Penelitian ini dilakukan karena kondisi jaringan internet yang tidak stabil di daerah perbatasan Entikong tetapi kebutuhan akan aplikasi dan kontrol dari pihak terkait sangat diperlukan, sehingga perlu dibuat suatu teknologi yang diharapkan dapat menghasilkan luaran berupa sebuah teknologi tepat guna yang dapat melakukan optimasi pada sebuah aplikasi berbasis *website* dengan kondisi jaringan internet yang *low bandwidth* bahkan dengan kondisi jaringan internet yang sering gangguan. Selain itu, penelitian ini diharapkan dapat menjadi suatu publikasi ilmiah di jurnal yang menjelaskan mengenai bagaimana penerapan *Progressive Web Apps* (PWA) dan *Load Balancing Server* pada aplikasi berbasis *website* dengan kondisi jaringan internet yang sering gangguan (tidak stabil).

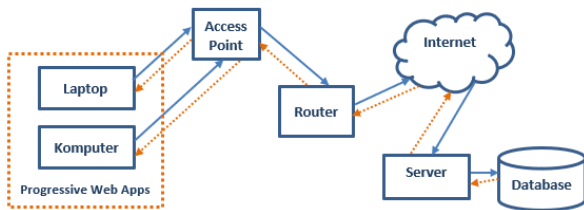
## 2. Metode Penelitian

Tahapan ini dilakukan dengan mencari berbagai sumber yang berhubungan dengan *Progressive Web Apps* (PWA), maupun optimasi terhadap aplikasi berbasis web melalui buku-buku dengan topik yang sesuai dengan penelitian ini, jurnal, paper, karya tulis ilmiah maupun literatur yang didapat dari internet. Selanjutnya, dilakukan proses penggalan informasi tentang kebutuhan sistem, seperti bahan dan peralatan apa saja dibutuhkan untuk membangun aplikasi dengan PWA. Informasi yang didapat akan menentukan seperti apa optimasi aplikasi yang akan dilakukan dan merupakan bagian penting dalam membangun sebuah aplikasi. Ada dua cara yang dapat dilakukan pada tahap analisis kebutuhan, yaitu dapat dilakukan dengan

metode *Waterfall* atau dilakukan dengan metode *Prototype*.

2.1 Rancangan Penelitian

Secara umum rancangan penelitian melibatkan infrastruktur dari sisi perangkat jaringan komputer, mulai dari penggunaan perangkat *access point* dan *router* yang terhubung ke internet. Setiap pengguna yang akan mengakses aplikasi dapat harus terkoneksi dengan internet. Pada kondisi aplikasi normal, saat adanya koneksi internet, maka akses aplikasi berjalan dengan lancar, tetapi ketika jaringan internet mengalami gangguan atau terputus, maka aplikasi atau *website* tidak dapat diakses kembali. Sedangkan dengan penerapan PWA halaman aplikasi atau *website* tetap bisa diakses seperti layaknya menggunakan koneksi internet, walaupun ketika dalam proses pengaksesan terjadi gangguan pada koneksi internet atau internet terputus. Untuk lebih jelasnya alur tersebut dapat dijelaskan seperti pada blok diagram Gambar 1.



Gambar 1. Blok Diagram Penelitian

Arsitektur PWA pada penelitian ini yaitu dengan adanya pemeran utama yang disebut dengan *Service Worker*. *Service Workers* adalah teknologi dibalik PWA yang memberikan fungsionalitas *offline*, notifikasi, update konten, pergantian konektivitas dan lainnya. Dengan konsep *offline first* dan Web APIs baru, aplikasi yang berjalan pada *browser* tidak akan memunculkan pesan network error atau white screen seperti yang selama ini seringkali dialami ketika berada pada jaringan yang sangat buruk. Dengan PWA, ketika kita meninggalkan *browser*, dan kembali lagi membuka aplikasi tersebut pada jaringan buruk atau tanpa jaringan sekalipun kita tidak akan kehilangan UI dan data-data terakhir. Untuk lebih jelasnya mekanisme *Service worker* PWA dapat dijelaskan pada Gambar 5.

2.2 Teknik Pengumpulan dan Analisis Data

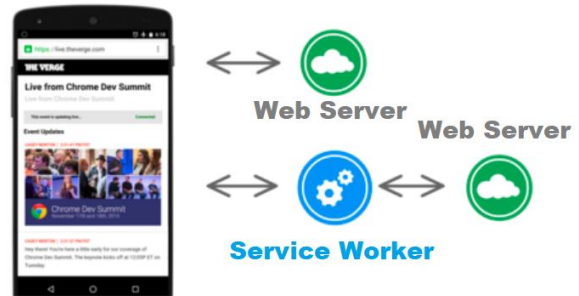
Data yang akan digunakan dalam penelitian ini adalah data *traffic* jaringan pada lokasi penelitian. Data-data *traffic* tersebut, akan dianalisa apakah ada yang dapat koneksi internet dilokasi penelitian mengalami gangguan atau tidak, sehingga dari hal tersebut dapat diketahui kapan PWA bekerja dengan baik dan stabil.

2.3 Rancangan Hak Akses

Pengguna sistem pada penelitian ini dibatasi oleh 3 jenis pengguna yaitu Operator (User 1) Admin (User 2)

dan Manager (User 3). Setiap pengguna memiliki tugas dan hak akses masing-masing.

Service Worker



Gambar 2. Mekanisme *Service worker* PWA

Tabel 1. Pembagian Akses User

User	Tugas
User 1	Pencarian Data Kendaraan. Mengelola Data Kendaraan Malaysia-Brunei masing-masing PLBN. Mengelola Data Kendaraan Indonesia masing-masing PLBN. Mengelola Data Kendaraan Tes masing-masing PLBN. Melihat Grafik perkembangan Tahun masing-masing PLBN.
User 2	Pencarian Data Kendaraan. Mengelola Keseluruhan Data Malaysia-Brunei setiap PLBN. Mengelola Keseluruhan Data Kendaraan Indonesia setiap PLBN. Mengelola Keseluruhan Data Kendaraan Tes setiap PLBN. Melihat Grafik Perkembangan Tahun disetiap PLBN.
User 3	Pencarian Data Kendaraan. Melihat Laporan Data Kendaraan Malaysia-Brunei. Melihat Laporan Data Kendaraan Indonesia. Melihat Laporan Data Kendaraan Tes. Melihat Grafik Perkembangan Tahun disetiap PLBN.

2.4 Kebutuhan Fungsional

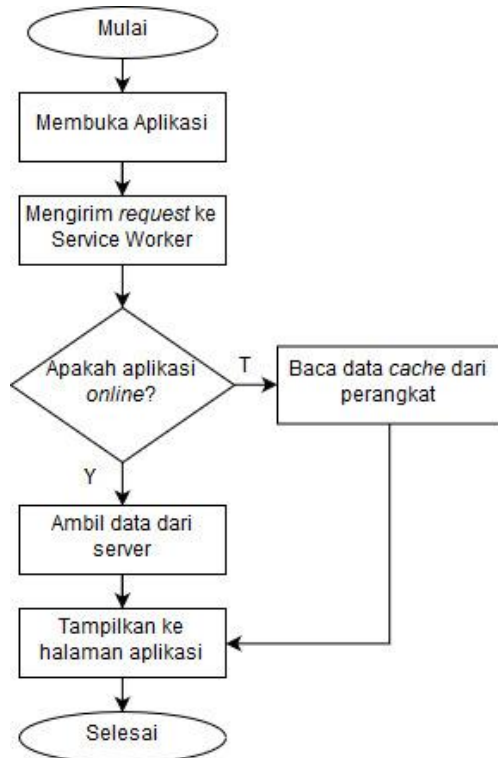
Aplikasi yang dibangun merupakan aplikasi pelayanan untuk mendata kendaraan yang melintasi daerah perbatasan Indonesia. Oleh karena itu, beberapa kebutuhan pada aplikasi ini berhubungan dengan kendaraan yang masuk dan keluar melewati perbatasan. Adapun kebutuhan fungsional adalah sebagai berikut:

1. Pencarian Data Kendaraan
2. Menambahkan Data Masuk Kendaraan dari Malaysia/Brunei
3. Menambahkan Data Keluar Kendaraan dari Malaysia/Brunei
4. Menambahkan Data Perpanjangan Kendaraan dari Malaysia/Brunei
5. Menambahkan Data VCC Kendaraan dari Malaysia/Brunei
6. Menambahkan Data Masuk Kendaraan Indonesia
7. Menambahkan Data Keluar Kendaraan Indonesia

8. Menambahkan Data Perpanjangan Kendaraan Indonesia
9. Menambahkan Data VCC Kendaraan Indonesia
10. Tampilan Grafik Laporan Kendaraan

2.5 Pengujian Sistem

Pengujian sistem akan dilakukan pada kondisi *online*, kemudian setelah beberapa halaman diakses, koneksi internet akan dimatikan, dan akan dilihat apakah aplikasi dapat bekerja dengan kondisi *offline* atau internet terjadi gangguan. Selain pengujian dari sisi perangkat lunak, pengujian juga akan dilakukan pada jaringan yang dikonfigurasi sehingga dapat diatur kondisi-kondisi tertentu. Adapun alur kerja *Service Worker* pada aplikasi penelitian ini dapat dilihat pada Gambar 3.



Gambar 3. Flowchart pengecekan halaman website

Hasil dari pengujian sistem, akan menentukan kelayakan sistem yang telah dibuat dapat diimplementasikan pada aplikasi pengawasan kendaraan lintas batas negara yang diakses oleh pengguna dari daerah perbatasan dengan kondisi internet yang sering mengalami gangguan atau dalam keadaan *offline*. Untuk implementasi dilakukan di daerah perbatasan yaitu PLBN Entikong, Kalimantan Barat.

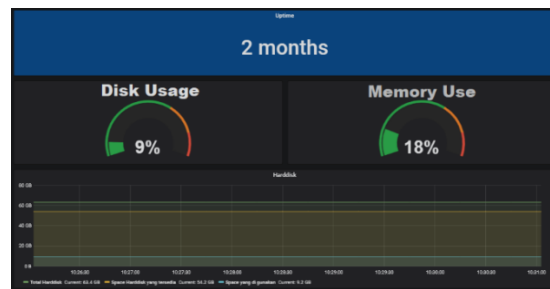
3. Hasil dan Pembahasan

3.1 Implementasi Server

Pada penelitian ini penerapan server dilakukan dengan mekanisme *load balancing*. Tujuan dari hal tersebut

agar server selalu dalam keadaan menyala dan terakses oleh pengguna dalam kondisi apapun. Adapun monitoring server dari aplikasi pengawasan kendaraan tersebut dapat dilihat pada Gambar 4.

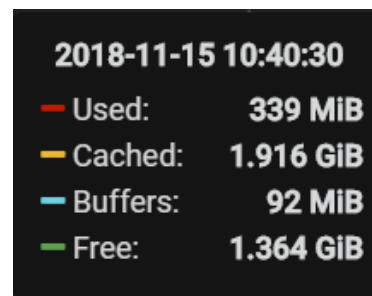
Pada Gambar 5 dapat dilihat kinerja memori server yang dikontrol setiap beberapa detik penggunaan dan pengaksesan aplikasi. Pada monitoring tersebut, terbagi menjadi beberapa kategori yang dapat di data, yaitu seberapa besar memori yang telah terpakai, memori *cache*, memori *buffer* dan sisa memori yang belum digunakan. Semua terpantau secara *real-time* sehingga dapat memudahkan administrator sistem dalam memantau kualitas memori server aplikasi tersebut.



Gambar 4. Monitoring Load Balancing Server



(a)

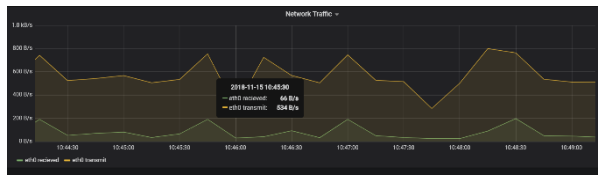


(b)

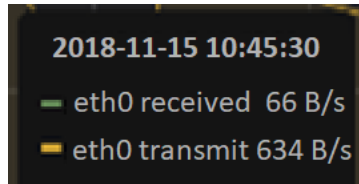
Gambar 5. Monitoring Memory Server (a) dan Indeks Penggunaan Memori (b)

Selain itu, pada server monitoring juga dilakukan pemantauan terhadap kondisi jaringan pengaksesan aplikasi dan server oleh pengguna. *Traffic* jaringan yang terpantau secara *realtime* sehingga dapat diketahui dengan cepat apabila ada serangan terhadap jaringan ke server. Selain itu monitoring ini bertujuan agar saat internet dalam keadaan down, aplikasi masih

dapat bekerja dengan metode PWA. Tampilan monitoring jaringan dapat dilihat pada Gambar 6.



(a)

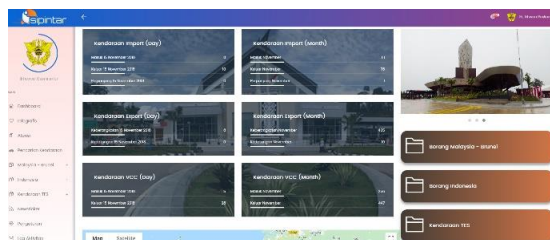


(b)

Gambar 6. Monitoring Jaringan Server (a) dan Indeks Kecepatan (b)

### 3.2 Implementasi Perangkat Lunak

Aplikasi pengawasan kendaraan di perbatasan dibuat menggunakan berbasis web, sehingga aplikasi hanya ditanam di server *online*, bisa diakses dimana saja, asalkan ada akses internet. Aplikasi tersebut terdiri dari 3 level akses, yaitu operator, admin dan manager. Gambar 7 merupakan tampilan utama dari aplikasi pengawasan kendaraan di daerah perbatasan.



(a)

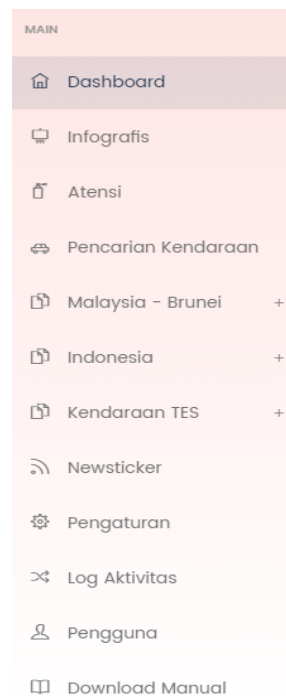


(b)

Gambar 7. Tampilan Utama Aplikasi Pengawasan Kendaraan (a) dan Layanan Informasi Kendaraan (b)

### 3.3 Fitur Menu Aplikasi

Pada penelitian ini, menu-menu yang dibuat pada aplikasi pengawasan kendaraan di perbatasan terdiri dari menu dashboard, yang berisi rangkuman jumlah data yang sudah diinputkan pada aplikasi, seperti data kendaraan masuk, kendaraan keluar, kendaraan vcc, perpanjang kendaraan baik dalam hari ini maupun bulan ini. Menu Info grafik berisikan tentang data statistik yang ditampilkan dalam bentuk grafik dari masing-masing kategori kendaraan, baik yang masuk ke Indonesia maupun yang keluar Indonesia. Selain itu, menu lainnya juga dapat melakukan pencarian kendaraan dari maupun yang keluar Indonesia. Menu tampilan data untuk kendaraan Indonesia maupun Malaysia Brunei dibuat terpisah, sehingga memudahkan dalam proses pengelolaan data maupun performa. Adapun fitur-fitur menu apa saja yang ada pada Aplikasi Pengawasan tersebut dapat dilihat pada Gambar 8.



Gambar 8. Menu-menu Aplikasi Pengawasan

### 4 Fitur Input Data

Ada beberapa fitur inputan data pada Aplikasi Pengawasan kendaraan yang telah dibuat. Fitur tersebut terdiri dari beberapa bagian, yaitu pada Borang Malaysia Brunei – Kendaraan Masuk, Keluar dll. Untuk lebih jelasnya, menu inputan data apa saja yang ada pada Aplikasi Pengawasan dapat dilihat pada Gambar 9.

### 5 Pencarian Kendaraan

Pada fitur pencarian kendaraan, operator maupun admin dapat melakukan pencarian kendaraan

berdasarkan dua (2) kategori, yaitu kendaraan Malaysia/Brunei dan kendaraan Indonesia. Pencarian dilakukan dengan memasukkan Nomor Polisi dari kendaraan yang akan dicari. Apabila kendaraan tersebut ditemukan maka, akan didapat detail dari kendaraan dan histori masuk dan keluar kendaraan tersebut di perbatasan. Untuk lebih jelasnya, hasil proses pencarian kendaraan dapat dilihat pada Gambar 10.



Gambar 9. Menu Fitur Input Data

Pencarian Data				
QBA7426				
Nomor Polisi	: QBA7426			
Nama Pemilik	: RUSLAN B JON			
Merk Kendaraan	: HELLX			
Warna Kendaraan	: PERAK			
Tahun Kendaraan	: 2010			
Nomor Chasis	: PM2M602S002146169			
Nomor Mesin	: T59A79T			
Riwayat Perjalanan				
No	Tgl Masuk	Tgl Perpanjang	Tgl Keluar	Nama Pengemudi
1	04 November 2018			RUSLAN B JON
2	25 Februari 2018		08 September 2018	RUSLAN B JON
3	21 Agustus 2017		04 September 2017	RUSLAN B JON
4	08 Juli 2017		08 Juli 2017	RUSLAN B JON
5	06 Juli 2017		07 Juli 2017	RUSLAN B JON
6	19 Juni 2017		20 Juni 2017	DEDI PURNAWAN SYARAH
7	20 Januari 2017		19 Februari 2017	RUSLAN B JON

Gambar 10. Tampilan Hasil Pencarian Kendaraan

## 6 Tabel Data Kendaraan

Untuk melihat kendaraan, pada aplikasi pengawasan kendaraan sudah disediakan fitur untuk melihat detail kendaraan sehingga dapat mengetahui informasi lebih jauh tentang kendaraan tersebut, untuk lebih jelasnya dapat dilihat pada Gambar 11 (a) dan (b).

Detail Kendaraan Masuk	
Masuk VIA	: Aruk
Nama Pemilik	: MOHAMAD ISZUAN BIN ROPEN
Nama Pengemudi	: ABANG SAUDI BIN ABANG MAN
Warga Negara	: MALAYSIA
No. SIM / Paspor	: K39180354
Tanggal Masuk	: 14 November 2018
Jam Masuk	: 16:13
Tanggal Perpanjangan	:
Tanggal Expired	: 14 Desember 2018
Merk Mobil	: PERODUA
Tahun	: 2014
Warna	: Putih
No. Polisi	: QAA2311J
No. Borang	: 840M/NOVEMBER/2018
No. Chasis	: PM2M602S002146169
No. Mesin	: T59A79T
Log Kendaraan	
<input type="button" value="Keluar"/>	

(a)



(b)

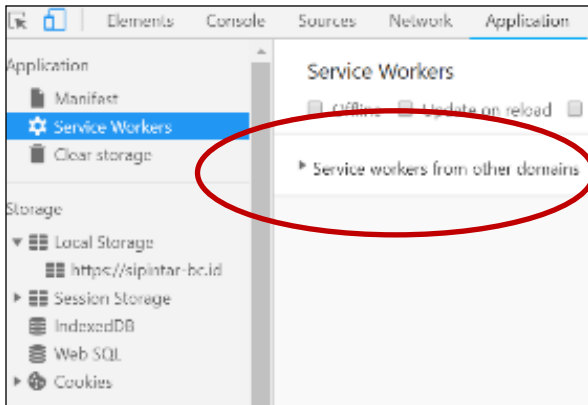
Gambar 11. (a) Informasi Detail Kendaraan (b) Foto Kendaraan

## 7 Implementasi Progressive Web Apps (PWA)

Aplikasi Pengawasan Kendaraan yang telah dibangun diakses di daerah perbatasan, sehingga memiliki keterbatasan untuk koneksi internet. Dalam kondisi seperti, penerapan PWA menjadi salah satu solusi untuk membuat website tetap dapat diakses walaupun dengan koneksi internet yang tidak stabil.

Pada penelitian ini, telah dibuat mekanisme PWA untuk mengakomodir pengaksesan aplikasi pengawasan tanpa memperhatikan performa koneksi internet yang stabil. Untuk mengetahui apakah *website* aplikasi sudah memiliki mekanisme PWA dapat melalui tab *service worker* yang ada pada *browser*. Pada aplikasi pengawasan kendaraan, dapat dilihat pada Gambar 12 telah terdeteksi adanya *service worker* yang bekerja

pada aplikasi tersebut, sehingga dapat dipastikan bahwa mekanisme PWA telah diterapkan.



Gambar 12. Hasil Pengecekan *Service Worker* pada Aplikasi

Proses adanya mekanisme PWA akan tercatat pada *Storage* lokal yang digunakan pada *browser*, sehingga dapat dilihat seberapa besar penggunaan *Storage* dari suatu aplikasi dengan mekanisme PWA. Gambar 13 menunjukkan kondisi *Storage browser* pada saat mengakses aplikasi Sipintar.

Path	Content-Type	Conte...	Time Cached
/	text/html; charset=UTF-8	2,674	4/2/2018, 3:19:54 PM
images/still_life_medium.jpg	image/jpeg	28,204	4/2/2018, 3:19:54 PM
index.html	text/html; charset=UTF-8	2,674	4/2/2018, 3:19:54 PM
pages/404.html	text/html; charset=UTF-8	1,080	4/2/2018, 3:19:54 PM
pages/offline.html	text/html; charset=UTF-8	1,121	4/2/2018, 3:19:54 PM
style/main.css	text/css; charset=UTF-8	2,771	4/2/2018, 3:19:54 PM

Gambar 13. Hasil Pengecekan *Storage* pada *Browser*

Untuk menyelesaikan salah satu permasalahan dengan koneksi internet yang dalam kondisi yang tidak stabil, maka pada aplikasi Sipintar telah diterapkan mekanisme PWA untuk membuat aplikasi Sipintar tetap dapat diakses dengan lancar di daerah perbatasan pada khususnya. Pada Gambar 14 dapat dilihat penerapan PWA pada aplikasi Sipintar yang telah dimodifikasi khusus untuk tetap dapat menampilkan data walaupun koneksi internet dalam kondisi *offline*.



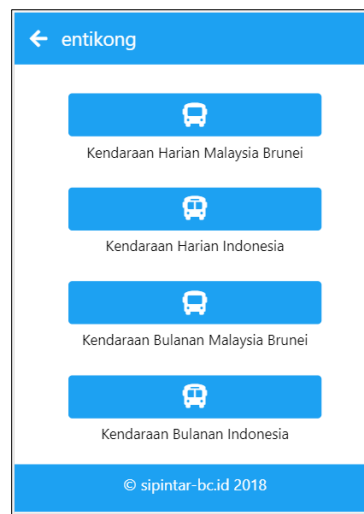
Gambar 14. Penerapan Identitas PWA pada Aplikasi Sipintar

Penerapan PWA dapat diketahui dari indikator pada tab *service worker* yang ada pada *browser*, dan meskipun internet dalam kondisi *offline*, tampilan website aplikasi tetap dapat diakses. Pengaruh pada penggunaan PWA pada aplikasi mengakibatkan adanya *storage* yang terpakai pada *browser*, sehingga *storage*

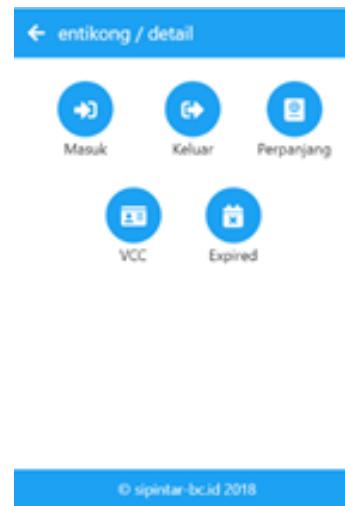
tersebut berfungsi sebagai cache apabila aplikasi diakses pada kondisi *offline*.

### 8 Antarmuka Aplikasi dengan Mekanisme PWA

Aplikasi Sipintar dengan penerapan PWA dibuat tidak serumit atau fitur-fiturnya selengkap aplikasi yang telah ada, akan tetapi hanya mengambil beberapa fitur informasi yang dibutuhkan, sehingga performa dari aplikasi tersebut tetap cepat dan mudah untuk diakses. Aplikasi Sipintar, telah diterapkan di tiga (3) daerah perbatasan, sehingga fitur menu yang diberikan dapat dipilih sesuai dengan daerah perbatasan yang diinginkan. Setelah menu daerah perbatasan dipilih, maka aplikasi dapat menampilkan data kendaraan harian dan bulanan baik untuk kendaraan Malaysia Brunei maupun kendaraan Indonesia, untuk lebih jelasnya dapat dilihat pada Gambar 15.



(a)



(b)

Gambar 15. Tampilan Data Kendaraan (a) dan Menu menggunakan PWA (b)

## 9 Pembahasan

Penelitian ini bertujuan untuk pengujian seberapa cepat respon suatu aplikasi web jika diterapkan *service worker* PWA pada saat keadaan *online* dan *offline*. Pada saat aplikasi pertama kali dijalankan secara *online* memerlukan waktu *transfer rate* yang agak lama karena *service worker* melakukan proses *catching* untuk pertama kali. Namun, ketika sistem dalam keadaan *offline* ataupun proses yang tidak memerlukan internet, aplikasi dapat bekerja secara cepat karena sistem hanya perlu memanggil *cache* yang sudah disimpan oleh *service worker*. Adapun *transfer rate* yang diperoleh saat proses *catching* oleh *service worker* dapat dilihat pada Tabel 2 dan 3.

Tabel 2. Kecepatan *Transfer Rate* Aplikasi saat keadaan *online*

Jumlah Data	Transfer Rate (detik)
10	3.4
50	3.65
100	5.57
200	9.5

Tabel 3. Kecepatan *Transfer Rate* Aplikasi saat keadaan *offline*

Jumlah Data	Transfer Rate (detik)
10	1.72
50	2.18
100	3.23
200	9.5

## 4. Kesimpulan

### 4.1 Simpulan

1. Penerapan PWA pada aplikasi sipinter, dapat bekerja meskipun kondisi internet tidak stabil, bahkan dalam kondisi *offline*, hal tersebut dapat dilihat pada proses pengujian pada saat *browser* dalam kondisi *offline*. Hanya saja pada saat kondisi *online*, kecepatan *transfer rate* lebih lama karena *service worker* sedang melakukan *caching* pada *browser*.
2. Tingkat keberhasilan dari penerapan PWA pada suatu aplikasi dapat dilihat pada indikator *manifest*, *service worker* dan *storage browser*. Pada aplikasi Sipinter ketiga hal tersebut telah terpenuhi, maka PWA sudah berhasil diterapkan.
3. *Load balancing* dilakukan pada penelitian ini digunakan untuk melayani apabila *traffic* jaringan penuh, dari pengujian server, aplikasi Sipinter hanya diakses oleh pengguna yang berada di lokasi Kalimantan Barat saja, sehingga mekanisme *Load balancing* belum begitu maksimal, dikarenakan *traffic* jaringan yang didapat pada saat pengujian hanya menghabiskan pemakaian tertinggi 1,3 Mbps.

### 4.2 Saran

1. Penerapan PWA cukup besar memakan sumber daya memori sehingga, perlu dipertimbangkan dalam menggunakan mekanisme PWA, jangan sampai sumber daya yang dimasukkan terlalu besar akan menghambat dalam pengaksesan aplikasi.
2. Pada penelitian selanjutnya dapat diterapkan dengan teknologi aplikasi berbasis *mobile* (android maupun ios) sehingga pengguna lebih memudahkan dalam mengakses aplikasi karena di perbatasan dukungan jaringan dari pihak penyedia jaringan GSM lebih stabil dibandingkan penyedia layanan internet berbasis industri rumahan.
3. Bagian terdiri atas simpulan dan saran atas penelitian hasil penelitian.

## Daftar Rujukan

- [1] A. Kurniawan, I. S. Areni dan A. Achmad, "Implementasi Progressive Web Application pada Sistem Monitoring Keluhan Sampah Kota Makasar," *JPE-Unhas*, vol. 21, no. 2, pp. 34-38, 2017.
- [2] S. Aghaei, "Evolution of the World Wide Web: From Web 1.0 to Web 4.0," *International Journal of Web & Semantic Technology*, vol. 3, p. 10, 2012.
- [3] M. R. Ridho, A. Pinandito dan R. K. Dewi, "Perbandingan Progressive Web Apps dan Mobile Web Terkait Waktu Respon, Penggunaan Memori dan Penggunaan Media Penyimpanan," *JPTIHK*, vol. 2, no. 10, pp. 3483-3491, 2018.
- [4] A. Sirajuddin dan E. Setijadi, "Rancang Bangun Server Learning Management System menggunakan Load Balancer dan Reverse Proxy," *Jurnal Teknik ITS*, Vol. 11 dari 12, no. 1, pp. 50-52, 2012.
- [5] A. Solichin, "Prinsip dan Cara Kerja Web Server," 2008. [Online]. Available: <http://achmatim.net/2008/07/09/prinsip-dan-cara-kerja-web-server/>. [Diakses 8 Mei 2017].
- [6] Y. Purwodiyogo dan A. Subardono, "Kontrol Bandwidth Dinamis Berbasis Algoritma Logika Fuzzy pada Jaringan Wireless Ad-Hoc," *Rekayasa Sistem dan Teknologi Informasi (RESTI)*, vol. 2, no. 3, 2018.
- [7] V. Karpagam, R. Padmavathe, R. Lakshana dan S. Priyadharshini, "Performance Enhancement of Webpage Using Progressive Web App Features," *International Journal of Innovative Research in Advanced Engineering (IJIRAE)*, vol. 4, pp. 97-103, 2017.
- [8] R. Julianto, W. Yahya dan S. R. Akbar, "Implementasi Load Balancing Di Web Server Menggunakan Metode Berbasis Sumber Daya CPU Pada Software Defined Networking," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 01, no. 09, pp. 904-914, 2017.
- [9] W. Li dan L. F. Kwok, "A Survey on OpenFlow-based Software Defined Network: Security Challenges and Countermeasures," *Journal of Network and Computer Applications*, vol. 68, pp. 126-139, 2016.
- [10] seon.co.id, "SSL dan HTTPS," 2018. [Online]. Available: <https://seon.co.id>. [Diakses Selasa April 2018].