

# Implementasi *Progressive Web Application* pada Sistem Monitoring Keluhan Sampah Kota Makassar

Awal Kurniawan<sup>\*1</sup>, Intan Sari Areni<sup>1</sup>, Andani Achmad<sup>1</sup>

<sup>1</sup>Teknik Elektro, Fakultas Teknik Universitas Hasanuddin

Jl. Perintis Kemerdekaan Km. 10, Tamalanrea Indah, Makassar,

\*Email: awal.kurniawan@uin-alauddin.ac.id

## Abstrak

Teknologi *web* sudah mengalami banyak kemajuan. Dimulai dari era *web 1.0* yang masih bersifat statis hingga teknologi *web* yang mampu mengatasi permasalahan perangkat keras seperti *storage*, *speech recognition*, hingga *geolocation*. Salah satu teknologi *web* yang hadir saat ini adalah *progressive web application*. Penelitian ini bertujuan untuk merancang sebuah sistem yang dapat melakukan proses *caching file* pada konten *website*. Sistem menggunakan *progressive web application* dengan memanfaatkan *service worker*. Sumber data yang akan dijadikan objek pada penelitian ini adalah data keluhan masyarakat yang berbentuk JSON. Pada penelitian ini digunakan metode eksperimental dalam merancang aplikasi. Data keluhan yang bersumber dari sebuah API (*Application Programming Interface*) kemudian ditampilkan dalam keadaan jaringan aktif. Selama dalam keadaan jaringan aktif, *service worker* melakukan tugasnya dalam melakukan proses *caching*. Setelah itu, data yang sudah disimpan bisa diakses pada jaringan tidak aktif. Hasil dari penelitian ini adalah sistem keluhan yang disisipkan *service worker* 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*.

## Abstract

Web technology has made great progress. Starting from the era of *web 1.0* that is still static to web technology that can solve hardware problems such as *storage*, *speech recognition*, to *geolocation*. One of the web technologies that are present today is *progressive web application*. This research aim to design a system that can perform the process a *caching file* on the *website* content. The system use *progressive web application* by utilizing *service worker*. The data source that will be used as object in this research is data of public complaints in the JSON form. In this research used experimental method in design of application. The complaint data sourced form in API (*Application Programming Interface*) is then displayed in an active network. As long as the network is active, the *service worker* does its work in the process of *caching*. After that, the data already saved can be accessed on the offline. The result of this research is a system of complaints inserted *service worker* able to process data *caching* up to 500 data complaints. Although the execution time required in accessing the application is longer due to the installation of *service worker*, but application are accessed faster when in an offline state because the data is loaded in the *service worker* cache.

**Kata-kunci:** *Progressive Web Application, Service Worker, Cache, Data Keluhan*

## 1. Pendahuluan

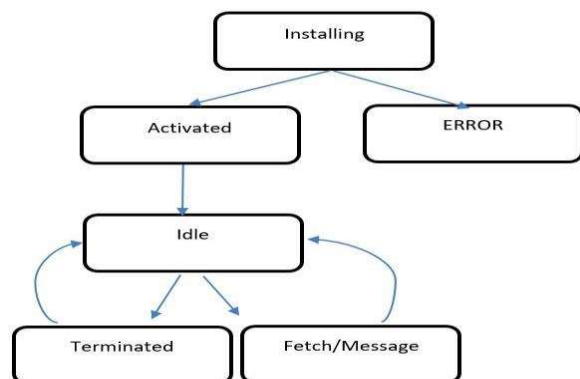
Saat ini teknologi aplikasi *web* sudah banyak mengalami perubahan fungsi. Awal mula perkembangan teknologi *web* dimulai dari *web 1.0* yang diperkenalkan tahun 1990 yang masih bersifat statis hingga menjadi aplikasi *web* yang dapat menangani masalah pengecekan status baterai, penggunaan *mode offline*, hingga *speech recognition* [1]. Salah satu teknologi yang tengah banyak diperhatikan saat ini adalah teknologi *Progressive Web App* (PWA).

*Progressive Web Apps* (PWA) adalah sebuah istilah untuk aplikasi berbasis *web* yang menggunakan teknologi *web* paling mutakhir. PWA sebenarnya hanyalah aplikasi berbasis *web* biasa, tapi memanfaatkan fitur perambanan yang modern agar tampil seolah-olah merupakan aplikasi asli. PWA digambarkan sebagai kumpulan dari teknologi, konsep desain dan *WEB API* (*Application Programming Interface*) yang bekerja secara bersama untuk memberikan sentuhan aplikasi pada sebuah *mobile web* [2]. Hal ini termasuk berbagai rekomendasi yang tidak spesifik pada desain aplikasi *web* untuk



perangkat mobile, seperti preferensi HTTPS melalui HTTP dan desain yang *responsive* [3]. Hal ini juga akan membawa kebutuhan pada API baru untuk peningkatan kualitas pengguna, seperti Web App Manifest, *Service Workers* ataupun *Payment Request API* [4].

Keuntungan dari PWA ini adalah dapat memuat konten seketika bahkan dalam kondisi jaringan yang tidak baik. Ketika digunakan dari layar awal pengguna, maka *service worker* diaktifkan pada PWA untuk dimuat seketika.



Gambar 1. Siklus *service worker*

Salah satu Konsep yang dibangun oleh PWA adalah *Service Worker*. *Service worker* adalah script yang berjalan di belakang *browser* pengguna. *Service worker* tidak membutuhkan sebuah halaman ataupun interaksi dari pengguna untuk menjalankan tugasnya, dengan begitu *service worker* akan terus berjalan walaupun halaman *web* tidak terbuka [5].

Langkah-langkah dalam melakukan konfigurasi dasar *service worker* sebagai berikut:

- Daftarkan *service worker* melalui URL (*Uniform Resource Locator*) fungsi *serviceWorkerContainer.register()*.
- Jika berhasil, *service worker* dijalankan di *ServiceWorkerGlobalScope*.
- *Service worker* telah siap untuk memproses *event*.
- Instalasi *service worker* dicoba ketika *service worker* mengontrol halaman yang diakses setelah dan sebelumnya. *Event install* akan selalu dikirim pertama kali ke *service worker*.
- Ketika handler *oninstall* selesai, *service worker* dipasang.
- Proses aktivasi. Ketika *service worker*

terpasang, selanjutnya akan menerima *event activate*. Penggunaan utama dari *onactivate* ini adalah untuk membersihkan sumber daya yang digunakan sebelumnya.

- *Service control* sekarang dapat mengontrol halaman, tapi hanya dibuka setelah *register()* telah sukses seperti dokumen mulai aktif dengan atau tanpa *service worker* dan menjaganya selama masih digunakan. Jadi dokumen harus dimuat ulang agar benar-benar terkontrol [6].

## 2. Bahan dan Metode

### 2.1. Lokasi dan Waktu Penelitian

Penelitian ini dilakukan di Lab Pusat Teknologi Informasi dan Pangkalan Data Universitas Islam Negeri Alauddin. Waktu Penelitian dilakukan selama 2 bulan yaitu bulan Mei 2017 sampai bulan Juli 2017.

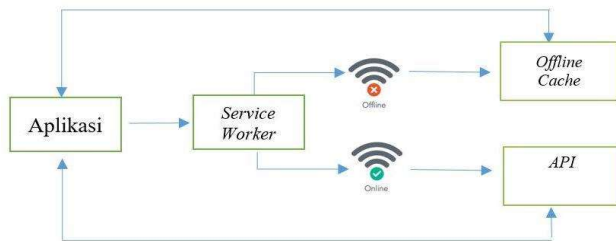
### 2.2. Sumber Data

Data yang digunakan pada penelitian ini adalah sebuah API yang berisi data keluhan sampah dari masyarakat yang terdiri dari foto keluhan, nama pelapor, tanggal melapor, serta deskripsi dari keluhan.

### 2.3. Rancangan Penelitian

Rancangan penelitian memiliki beberapa tahapan sesuai dengan metode *waterfall* [7]. Pertama, Studi literatur : pada tahap ini dilakukan dengan cara mengunjungi perpustakaan untuk mencari referensi berupa buku, tesis atau pun jurnal yang berhubungan dengan topik penelitian. Selain itu, studi literatur juga dilakukan dengan cara online atau mengunjungi suatu *website*, misalnya [www.google.com](http://www.google.com). Kedua, Analisis Kebutuhan Sistem: pada tahapan ini peneliti melakukan analisa terhadap sistem, apa saja yang menjadi kebutuhan sistem yang akan dirancang, seperti *software*, *hardware* atau sistem operasi, dan sebagainya. Dan untuk mendapatkan informasi tersebut, peneliti melakukan diskusi dan studi literatur. Ketiga, Desain Sistem: pada tahapan peneliti mendesain antarmuka aplikasi data keluhan untuk *client*. Keempat, Penulisan Kode Program : Kode program menggunakan bahasa pemrograman *TypeScript* dengan basis data

MongoDB yang berkonsep *NoSQL*. Dan yang terakhir, Pengujian Sistem: sistem diuji dengan menggunakan teknik pengujian *black box*. Setiap fungsional yang ada pada sistem diuji kinerjanya untuk mendapatkan sistem yang berkualitas [8].



Gambar 2. Rancangan sistem

#### 2.4. Metode Pengumpulan Data

Sumber data diperoleh dari sebuah API dengan alamat <https://last.uin-alauddin.ac.id/showAllKeluhan>. Data API terdiri dari `_id`, `namapelapor`, `gambar_keluhan`, `keterangan`, `koordinat` dan `tanggal_keluhan`.

#### 2.5. Analisis Data

Data yang didapatkan berupa file/assets yang dapat disimpan oleh *service worker* ketika aplikasi dijalankan. Data yang didapatkan bersumber dari sebuah API kemudian dianalisis

untuk membuktikan kinerja dari *service worker* dalam melakukan *caching file* data sehingga data keluhan yang diakses aplikasi tetap dapat dilihat walau dalam keadaan *offline*.

### 3. Hasil Penelitian

Pada proses antar muka sistem keluhan sampah ada beberapa pilihan yang mendukung proses kerja sistem. Secara singkat proses ini adalah sebagai berikut : Proses pengambilan data keluhan, Proses menyisipkan kode *service worker*, Proses menampilkan data keluhan di aplikasi dalam keadaan jaringan internet aktif, lalu proses menampilkan data keluhan di aplikasi dalam keadaan jaringan tidak aktif. Adapun tahapan-tahapan yang dilakukan dalam membangun aplikasi yaitu :

#### 3.1. Proses Pengambilan Data Keluhan

Adapun tahapan yang dilakukan dalam mengambil data keluhan dengan mengakses alamat API [9]. Bentuk data menggunakan format *JSON* yang terdiri dari `_id`, `namapelapor`, `gambar_keluhan`, `keterangan`, `koordinat` dan `tanggal_keluhan`.

```
[{"_id": "582bf8c80502bf3447f9b684", "username": "user30", "nama": "Nirwana", "fotoprofile": "user30.jpg", "fotokeluhan": "gambar30.jpg", "isikeluhan": "segera diangkut", "lat": "-5.150362", "lon": "119.431565", "v": 0, "create_date": "2016-11-16T06:12:24.815Z"}, {"_id": "582bf6fd0502bf3447f9b683", "username": "user29", "nama": "ipul", "fotoprofile": "user29.jpg", "fotokeluhan": "gambar29.jpg", "isikeluhan": "sudah lama tidak diangkut", "lat": "-5.158264", "lon": "119.410188", "v": 0, "create_date": "2016-11-16T06:04:45.402Z"}, {"_id": "582bf6150502bf3447f9b682", "username": "user28", "nama": "muzakkir", "fotoprofile": "user28.jpg", "fotokeluhan": "gambar28.jpg", "isikeluhan": "sudah lama tidak diangkut", "lat": "-5.152248", "lon": "119.424390", "v": 0, "create_date": "2016-11-16T06:00:53.492Z"}, {"_id": "582bf5770502bf3447f9b681", "username": "user27", "nama": "dg.farid", "fotoprofile": "user27.jpg", "fotokeluhan": "gambar27.jpg", "isikeluhan": "busuk", "lat": "-5.161181", "lon": "119.422942", "v": 0, "create_date": "2016-11-16T05:58:15.584Z"}, {"_id": "582bf54a0502bf3447f9b680", "username": "user26", "nama": "Reza M", "fotoprofile": "user26.jpg", "fotokeluhan": "gambar26.jpg", "isikeluhan": "harap segera sampahnya diangkut", "lat": "-5.159474", "lon": "119.434012", "v": 0, "create_date": "2016-11-16T05:57:30.595Z"}, {"_id": "582bf30c0502bf3447f9b67f", "username": "user25", "nama": "Syahbuddin", "fotoprofile": "user25.jpg", "fotokeluhan": "gambar25.jpg", "isikeluhan": "bau", "lat": "-5.154092", "lon": "119.440970", "v": 0, "create_date": "2016-11-16T05:47:56.656Z"}, {"_id": "582bf2e30502bf3447f9b67e", "username": "user24", "nama": "gunawan", "fotoprofile": "user24.jpg", "fotokeluhan": "gambar24.jpg", "isikeluhan": "bau sangat menyakkan", "lat": "-5.151965", "lon": "119.423324", "v": 0, "create_date": "2016-11-16T05:47:15.102Z"}, {"_id": "582bf05e0502bf3447f9b67d", "username": "user23", "nama": "Asep Indra", "fotoprofile": "user23.jpg", "fotokeluhan": "gambar23.jpg", "isikeluhan": "Semoga cepat dibersihkan", "lat": "-5.153771", "lon": "119.448126", "v": 0, "create_date": "2016-11-16T05:36:30.561Z"}, {"_id": "582bf0360502bf3447f9b67c", "username": "user22", "nama": "Ridwan", "fotoprofile": "user22.jpg", "fotokeluhan": "gambar22.jpg", "isikeluhan": "Semoga cepat dibersihkan", "lat": "-5.135485", "lon": "119.415852", "v": 0, "create_date": "2016-11-16T05:35:50.607Z"}]
```

Gambar 3. API isi keluhan dalam format json

#### 3.2. Proses Penyisipan Kode *Service Worker*

Proses penyisipan kode *service worker* dilakukan dengan menyisipkan kode ke dalam file utama dalam aplikasi. Ketika kode berhasil dipasang, maka proses *service worker* akan bekerja dalam melakukan proses *caching* data.

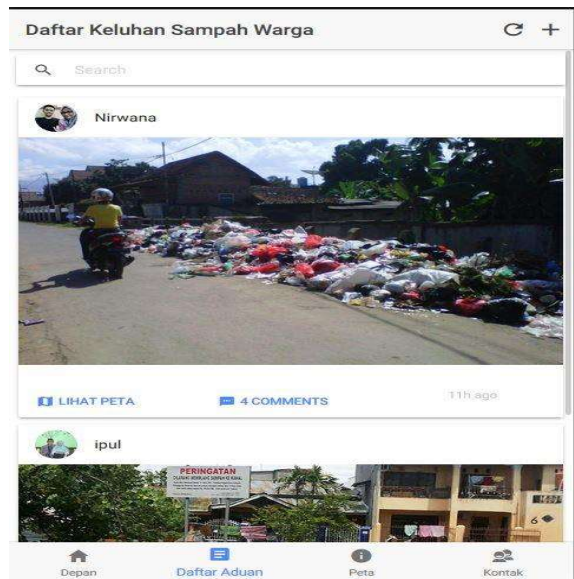
```
<script>
if ('serviceWorker' in navigator) {
  navigator.serviceWorker.register('service-worker.js')
    .then(() => console.log('service worker installed'))
    .catch(err => console.error('Error', err));
}
</script>
```

Gambar 4. Pemasangan kode *service worker*



### 3.3. Proses Pembacaan Data Keluhan Pada Jaringan Aktif

Proses pembacaan data diawali dengan client melakukan request data keluhan ke server. Sistem yang terdapat di server kemudian mengirimkan *response* berupa data keluhan ke aplikasi. Sistem akan membandingkan data pesanan dari *client* dengan data yang sudah ada di server. Ketika jaringan dalam keadaan aktif, maka aplikasi akan melakukan proses *caching file* dengan menyimpan data-data keluhan seperti nama pelapor, gambar keluhan, tanggal aduan, koordinat, dan keterangan keluhan.



Gambar 5. Tampilan aplikasi daftar keluhan

### 3.4. Proses Pembacaan Data Keluhan Pada Jaringan Tidak Aktif

Proses pembacaan data keluhan pada jaringan tidak aktif dilakukan dengan menonaktifkan jaringan terlebih dahulu. Hasil data yang dilakukan pada pembacaan data jaringan yang tidak aktif kemudian dibandingkan dengan hasil data pada jaringan yang aktif. Proses analisis hasil data akan dilakukan dalam jumlah aduan yang berbeda, yaitu 5, 10, 25, dan 100 data.

## 4. Pembahasan

Penelitian ini memperlihatkan hasil dari proses *caching* yang dilakukan oleh *service worker*. Dari hasil tabel, dapat dilihat ketika aplikasi pertama kali dijalankan secara *online*,

maka akan memerlukan proses transfer data lebih lama karena sistem melakukan proses *caching* dari *service worker*. Ketika jaringan dalam keadaan *offline*, aplikasi dapat diakses dengan cepat karena hanya memuat data yang sudah disimpan oleh *service worker*.

Tabel 1. Proses *caching file* oleh *service worker* dalam jaringan aktif (*online*)

Jumlah Data	Transfer Rate	Transfer Data
10	3.6 detik	52.5 kb
50	3.74 detik	100 kb
100	6.18 detik	1100 kb
500	27.8 detik	21400 kb

Tabel 2. Hasil proses *caching file* oleh *service worker* dalam jaringan tidak aktif (*offline*)

Jumlah Data	Transfer Rate	Transfer Data
10	2.20 detik	37.4 kb
50	2.81 detik	37.4 kb
100	3.98 detik	37.4 kb
500	21.15 detik	37.4 kb

## 5. Kesimpulan

Pemanfaatan PWA dalam aplikasi web dapat menjadi solusi bagi pengguna aplikasi. Dengan memanfaatkan fitur *service worker* yang mampu melakukan *caching file* hingga ratusan data keluhan, sehingga aplikasi tetap dapat dijalankan meskipun dalam keadaan *offline*. Meskipun eksekusi waktu yang dibutuhkan dalam mengakses aplikasi lebih lama karena pemasangan *service worker*, namun aplikasi yang diakses pun lebih cepat dalam memuat data keluhan warga karena data diakses dari *cache service worker*.

## Ucapan Terimakasih

Penulis menyadari bahwa penyusunan penelitian ini banyak mengalami hambatan, rintangan dan halangan, namun dengan bantuan dari berbagai pihak semua ini dapat terselesaikan dengan baik. Oleh karena itu, pada kesempatan ini penulis menyampaikan rasa terima kasih dan penghargaan yang setinggi-tingginya kepada Ibu Intan Sari Areni serta Andani Achmad selaku pembimbing yang banyak meluangkan waktunya memberikan petunjuk dan bimbingan sehingga

kesulitan penulis dalam membuat tesis ini dapat terselesaikan. Terima kasih juga yang sebesar-besarnya kepada Bapak dan Ibu Dosen yang telah memberikan banyak masukan dan bimbingannya selama penulis menempuh perkuliahan.

## Referensi

- [1] What Web Can Do Today, "What Web Can Do Today", diunduh [4 juli 2017] melalui <https://whatwebcando.today>
- [2] Rahul Surendra Mishra, "Progressive WEBAPP : Review", International Research Journal Of Engineering and Technology (IRJET), Vol.3, Issue 6, 2016.
- [3] Google Developer, "Introduction to Service Worker", diunduh [4 Juli 2017] melalui <https://developers.google.com/web/ilt/pwa/introduction-to-service-worker/>
- [4] Mozilla Developer, "Service Worker API", diunduh [4 Juli 2017] melalui [https://developer.mozilla.org/en-US/docs/Web/API/Service\\_Worker\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Service_Worker_API)
- [5] Google Developer, "Progressive Web App Checklist", diunduh [4 Juli 2017] melalui <https://developers.google.com/web/progressive-web-apps/checklist>
- [6] Mozilla Developer, "Using Service Worker", diunduh [4 Juli 2017] melalui [https://developer.mozilla.org/en-US/docs/Web/API/Service\\_Worker\\_API/Using\\_Service\\_Worker](https://developer.mozilla.org/en-US/docs/Web/API/Service_Worker_API/Using_Service_Worker)
- [7] Pressman Roger S., "Rekayasa Perangkat Lunak", Penerbit Andi:Yogyakarta
- [8] Pressman Roger S. (2002). Rekayasa Perangkat Lunak Pendekatan Praktisi. Penerbit Andi:Yogyakarta.
- [9] Muhsin Shodiq, M. dkk. 2015. Implementation of Data Synchronization with Data Marker using Web Service Data. Bina Nusantara University, Palmerah Jakarta Barat, Indonesia.

