

IMPLEMENTASI PROGRESSIVE WEB APPLICATION PADA SISTEM INFORMASI PASAR LELANG KOMODITI PERTANIAN

Erfanti Fatkhiyah¹, Joko Triyono^{2*}, Haidar Ikram Ramadhan³, Nizar Izzuddin Yatim Fadlan⁴

^{1,3,4}Informatika IST AKPRIND Yogyakarta, ²Rekayasa Sistem Komputer IST AKPRIND Yogyakarta
e-mail : ¹erfanti@akprind.ac.id, ²jack@akprind.ac.id, ³daihard.haidar@gmail.com, ⁴fadlannizar5@gmail.com
*Corresponding Author: jack@akprind.ac.id

ABSTRACT

Technology always has an impact on how products and services are designed. In recent years, there has been unprecedented growth in the web field. Therefore, it requires fast, reliable, attractive development and powerful applications. One of them is a product from Google, namely google sheets. The results of the data can be presented in full either in the form of sheets or graphic information, so that managers or admins do not need to process anymore and later will be able to be monitored on a web-based app system that uses Progressive Web Application (PWA) technology. In the development of information needs, it turns out that information needs have become very complex both online and offline, even today's technology can combine the transaction process on the local information system synchronized to the information system and or online data. The implementation of Progressive Web Application (PWA) makes it possible to distribute information more easily. The results of this research obtained modeling, designing and implementing local applications with data synchronized to google sheet media, data on google sheets will be accessed by different applications online using Progressive Web Application (PWA). it is also proven that with the implementation of Progressive Web Application (PWA), data after being uploaded and cached can still be accessed even if the internet communication line is interrupted. Because the application will always read the cache that has contained data as long as it is not synchronized data.

Keywords : *Distribustion, Google, JSON, PWA*

INTISARI

Teknologi selalu berdampak pada bagaimana produk dan layanan dirancang. Dalam beberapa tahun terakhir, telah terjadi pertumbuhan yang belum pernah terjadi sebelumnya di bidang web. Oleh karena itu, diperlukan pengembangan yang cepat, andal, menarik, dan aplikasi yang kuat. Salah satunya produk dari google yaitu google sheet. Hasil data dapat tersaji secara lengkap baik berupa sheet ataupun informasi grafis, sehingga pengelola atau admin tidak perlu untuk mengolah lagi dan nantinya akan dapat dimonitoring pada sistem app berbasis web yang menggunakan teknologi Progressive Web Apps (PWA). Dalam perkembangan kebutuhan informasi, ternyata kebutuhan informasi menjadi sangat komplek baik online maupun offline, bahkan teknologi saat ini sudah bisa menggabungkan proses transaksi pada sistem informasi lokal tersinkronisasi ke sistem informasi dan atau data online. Penerapan Progressive Web Apps (PWA) memungkinkan untuk mendistribusikan informasi secara lebih mudah. Hasil Penelitian ini diperoleh pemodelan, perancangan dan implementasi aplikasi lokal dengan data di sinkronisasi ke media google sheet, data pada google sheet akan diakses oleh aplikasi yang berbeda secara online menggunakan Progressive Web Application (PWA), terbukti juga bahwa dengan penerapan Progressive Web Application (PWA), maka data setelah diload dan masuk cache tetap bisa diakses walapun jalur komunikasi internet terputus. Karena aplikasi akan selalu membaca cache yang telah berisi data selama tidak dilakukan sinkronisasi data.

Kata kunci : *Distribusi, Google, JSON, PWA*

1. PENDAHULUAN

Penerapan teknologi informasi saat ini sudah memasuki semua sendi kehidupan bisnis maupun sosial kemasyarakatan. Kemudahan penggunaan *form online* menjadikan hal tersebut sebagai pilihan utama dalam membuat transaksi dan pelaporan karena memiliki fitur-fitur yang lengkap. Secara umum, masyarakat bisa diibaratkan seperti mesin pencari Google atau media sosial, karena setiap terdapat kegiatan yang berhubungan

dengan kegiatan pendataan maka hampir secara keseluruhan akan memanfaatkan formulit online.

Salah satunya produk dari *google* yaitu *google sheet*. Hasil pendataan dapat tersaji secara cepat dan lengkap mulai dari sheet sampai informasi grafis, sehingga pengelola atau admin tidak perlu melakukan proses lagi dan nantinya akan dapat dimonitoring pada sistem *app* berbasis *web* yang menggunakan teknologi *Progressive Web Apps (PWA)*. Dalam perkembangan kebutuhan informasi, ternyata kebutuhan informasi menjadi sangat kompleks baik online maupun offline, bahkan teknologi saat ini sudah bisa menggabungkan proses transaksi pada sistem informasi lokal tersinkronisasi ke sistem informasi dan atau data online. Penerapan *Progressive Web Apps* memungkinkan untuk mendistribusikan informasi secara lebih mudah.

Fasilitas *Google APIs*, *Share to Sheet*, *CSV* dll yang disediakan oleh *Google Sheet* dapat digunakan untuk memberikan informasi kepada pengguna lain. Selain itu, data dari *Google Form* juga dapat diakses oleh pengguna lain dengan menggunakan metode tertentu tanpa perlu login ke akun *Google Form*. Dukungan *SSO (Single Sign On)* untuk penyajian informasi dapat digunakan untuk berbagi informasi kepada pengguna lain di banyak platform, sehingga dengan konsep berbagi ini maka pengguna dapat berbagi informasi tanpa harus mengakses langsung dari penyedia informasi.

Hasil penelitian ini melakukan pemodelan, perancangan dan implementasi mulai dari aplikasi lokal yang datanya akan di sinkronisasi ke media *google sheet*, untuk selanjutnya data pada *google sheet* akan diakses oleh aplikasi yang berbeda secara online menggunakan *PWA*. Pada penelitian ini akan diangkat judul Implementasi *Progressive Web Application* Pada Sistem Informasi Pasar Lelang Komoditi Pertanian.

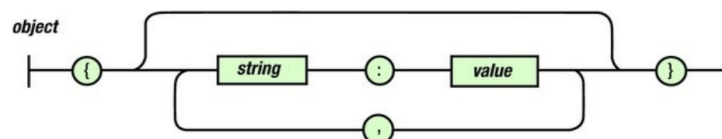
Hasil dari penelitian yang dilakukan oleh (Triyono & Fatkhiyah, & Wicaksono, 2021) didapat kesimpulan bahwa dengan memanfaatkan file teks untuk menampung data terdistribusi yang diperoleh dari sebuah *APIs* menjadikan penggunaan bandwidth bisa efektif dan efisien serta respon time aplikasi meningkat, aplikasi yang menerapkan teknik ini mendapatkan keunggulan lain yaitu bisa berjalan dengan baik, tanpa selalu harus terkoneksi dengan penyedia data utama. Hal ini terjadi karena data sudah tersimpan secara lokal pada masing-masing sistem informasi.

Aplikasi *web front end* dapat mengimplementasikan teknik *JSON* (Triyono, Haryani, & Farid AS, Membangun Aplikasi *Web Front End* dengan Teknik *JSON (Java Script Object Notation)*, 2018) sehingga dapat disimpulkan bahwa *JSON* dapat dijadikan sebagai sebuah paradigma *DBMS (Database Management System)* sebagai salah satu teknik pengelolaan data. Adanya kecepatan akses ini menjadi salah satu kelebihan dari *JSON*, sehingga dapat dikombinasikan dengan penerapan *local storage* (penyimpanan lokal) pada *HTML5*. Perkembangan teknologi tersebut tentunya akan dapat memberikan dampak pada kebutuhan informasi sehingga memerlukan penambahan fasilitas-fasilitas dan modul-modul yang dapat dijalankan prosesnya tanpa mengganggu sistem informasi yang sedang berjalan. Pembagian sistem secara modular juga perlu mendapatkan perhatian dalam pengembangan sistem informasi sehingga sistem dapat bekerja secara bersamaan tanpa bergantung pada kondisi geografis dan lokasi. Teknik pengembangan *web front end* dapat menggunakan *JSON* untuk penyedia data, dan digunakan bersama dengan *HTML5* dan *PHP* dengan menerapkannya pada *server side with database* dan menghubungkan antar sistem. Ini akan menjadi luaran untuk penelitian ini. Hasilnya nanti informasi pada aplikasi pada sisi *client* dapat berjalan tanpa terkoneksi secara langsung ke *database*. Pengembangan *client* dapat dibuat berbasis web dengan *PHP*, membuat aplikasi berbasis *desktop*, ataupun membuat aplikasi *Android*.

Penelitian oleh (Triyono 2015) yaitu berkaitan dengan pengembangan prototipe sistem informasi yang dilengkapi dengan adanya jejaring sosial *twitter* yang dapat digunakan oleh petani dalam melaporkan kegiatan sehari-hari yang dilakukan ke sistem dengan fasilitas *APIs (Application Programming Interface)*. API ini dapat menerima informasi dan akan dikirimkan ke sistem menggunakan akun *twitter* milik pengirim. Dari sisi teknologi dan biaya petani, metode ini dapat dinilai baik karena mudah dijalankan. Selain itu, dari sisi investor akan mendapatkan kemudahan untuk melihat perkembangan hasil investasinya. Kelebihan lainnya adalah kepuasan pengguna website dapat disimpulkan bahwa dari uji hipotesis memperlihatkan beberapa dimensi seperti adanya *citizen support*, adanya *content and appearance of information*, adanya *functionality of the interaction environment*, *trust*, dan *easy of use* dapat memberikan dampak terhadap kualitas layanan *e-government* yang ada. Tingkat kepuasan dan intensitas pengguna sebagai kualitas layanan *e-government* juga diterapkan pada website Pemkot (Pemerintah Kota) Yogyakarta, sehingga dapat disimpulkan bahwa metode pengiriman informasi tak terbatas hanya pada website, tetapi juga bisa diterapkan untuk aplikasi yang berbeda-beda. Website dapat digunakan sebagai media untuk menampung informasi utama dan dilengkapi dengan media aplikasi lain untuk penyampaian informasi dan juga *notifikasi* untuk pengguna terdaftar atau dapat melakukan *broadcast* ke calon pengguna baru.

Menurut penjelasan dalam buku (Marrs 2017) yang berjudul *JSON at Work* dikatakan bahwa JSON telah menjadi standar *de facto* bagi pengembangan antarmuka dengan *RESTful*, walaupun beberapa bagian di dalamnya belum banyak diketahui sebagai standar, alat, dan juga teknologi yang digunakan oleh perancang dan pengembang pada saat pengembangan aplikasi. Fungsi *JSON* sebagai pengganti bagi XML saat melakukan panggilan pada *AJAX*. Fungsi ini sangat penting untuk pertukaran data menggunakan media *internet*. Standar dan implementasi terbaik dapat diterapkan dengan memanfaatkan kemampuan *JSON* dalam pembangunan aplikasi yang efisien, elegan dan berguna.

Struktur komposit *JSON* menjelaskan bahwa asal mula *JSON* berasal dari standarisasi *ECMAScript* (Smith 2015). Implementasi dua struktur dapat diwakili dalam format objek dan juga format larik. Kedua representasi struktural dari *JSON* tersebut digambarkan pada Gambar 1 dalam serangkaian diagram sintaks yang berisi representasi gramatikal dari pasangan string/nilai dalam pembangunan *JSON*.



Gambar 1. Sintak diagram dari koleksi pasangan string / nilai

Aplikasi di deskripsikan oleh sebuah model, Model tidak memperhatikan presentasi atau *user interface (UI)*, model di implementasikan dengan *JSON*.

```
{ "pegawai": [
  { "namadepan": "Joko", "namabelakang": "Triyono" },
  { "namadepan": "Endang", "namabelakang": "Susanti" },
  { "namadepan": "Bambang", "namabelakang": "Cahyo" }
]}
```

JSON dalam PHP

Program PHP mempunyai banyak fungsi bawaan dalam penanganan JSON. Adapun objek-objek dalam PHP

dapat diubah menjadi format JSON dengan menggunakan fungsi yang ada dalam PHP yaitu `json_encode ()`:
(Lorenz, 2022)

```
<?php
$myObj->nama = "Joko";
$myObj->umur = 55;
$myObj->kota = "Sleman";
$myJSON = json_encode($myObj);
echo $myJSON;
?>
```

Script diatas akan menghasilkan output sebagai berikut:

```
{"nama": "Joko", "umur": 55, "kota": "Sleman"}
```

Membuat struktur tampilan pada PWA

Web App Manifest adalah file JSON sederhana yang berisi informasi: nama, nama_pendek, deskripsi, ikon untuk resolusi perangkat yang berbeda, start_url, mode tampilan, warna tema aplikasi. Penggunaan Manifest Aplikasi Web menginstal aplikasi web di layar beranda pengguna antara aplikasi asli. Hasilnya, pengguna bisa mendapatkan akses cepat dan menikmati layar penuh tampilan seperti dengan aplikasi asli. (Szkarski, M. 2019)

Script manifest.json

```
{
  "name": "nama aplikasi",
  "short-name": "nama pendek dari aplikasi",
  "start_url": "/index.html",
  "scope": "/",
  "display": "standalone",
  "background_color": "#fff",
  "theme_color": "#3A1AB1",
  "description": "deskripsi dari apps",
  "dir": "ltr",
  "lang": "en-US",
  "orientation": "portrait-primary",
  "icons": [
    {
      "src": "/image/icon-192x192.png",
      "type": "image/jpg",
      "size": "192x192"
    },
    {
      "src": "/image/icon-512x512.png",
      "type": "image/jpg",
      "size": "512x512"
    }
  ],
  "prefer_related_applications": "true",
  "related_applications": [
    {
      "platform": "play",
      "url": "https://play.google.com/store/.....",
      "id": "com.example.app"
    }
  ]
}
```

Mengimplementasikan Web Manifest

Pada tulisan di atas berisikan perintah untuk membuat struktur tampilan pada *PWA*, ini harus diimplementasikan pada meta tag html untuk browser dapat membaca dan memahami sebagai sebuah konfigurasi tampilan *PWA*. (Szkarski, M. 2019)

Contoh script meta tag html:

```
<link rel="manifest" href="/manifest.json">
```

Membuat Service Worker

Dengan service worker dapat menginstall PWA dengan memasukkan url yang akan dimasukkan ke *Cache* dengan menambahkan list array url tersebut pada function *addAll(url)*. Secara otomatis service worker akan menyimpan file atau source code url tersebut pada *Cache*. (Szkarski, M. 2019)

Script service worker:

```
const urlsToCache = [ "/", "/style.css", "/script.js", "/data-petani.json",  
"/data-lelang.json" ];  
self.addEventListener("install", function(event) {  
  event.waitUntil(  
    caches.open("pwa").then(function(cache) {  
      return cache.addAll(urlsToCache);  
    })  
  );  
});  
self.addEventListener("fetch", function(event) {  
  event.respondWith(  
    caches.open("pwa").then(function(cache) {  
      return cache.match(event.request).then(function(response) {  
        if(response) {  
          return response;  
        }  
        return fetch(event.request);  
      });  
    })  
  );  
});
```

Mendaftarkan Service Worker

Service worker yang telah dibuat pada tulisan di atas harus didaftarkan pada browser dengan script *Javascript* dengan *navigator*. Cara mendaftarkan sebuah service worker pada *navigator* dengan function *serviceWorker.register(url: tempat service worker)* dengan deklarasi tersebut browser akan mendaftarkan file JavaScript yang berisi service worker. (Szkarski, M. 2019)

Script mendaftarkan service worker pada browser:

```
if ("serviceWorker" in navigator) {  
  navigator.serviceWorker.register("/sw.js")  
  .then(() => {  
    console.log("Service Worker is registered");  
  })  
  .catch(err => {  
    console.error("Registration failed:", err);  
  });  
}
```

2. METODE PENELITIAN

Uji coba dalam penelitian ini dilakukan dalam skala laboratorium dan dilakukan pada Laboratorium Jaringan dan Internet Institut Sains dan Teknologi AKPRIND Yogyakarta. Data atau model yang dipakai pada penelitian ini merupakan data transaksi pasar lelang Sendangtirto Kelurahan Sendangtirto tahun 2022 dan di simulasikan di laboratorium dan sekaligus diimplementasikan dalam sistem *online*. Batasan aplikasi dalam penelitian ini berfokus pada aplikasi yaitu dari sisi pemakai/petani pada tingkat menampilkan data dan atau mencari data yang bersumber dari *google sheet* yang real time secara online. *Google Sheet* dalam hal ini adalah data sinkronisasi hasil transaksi dari Aplikasi Pos Lelang, data tersebut akan disinkronkan setiap selesai jadwal

penerimaan lelang. **Aplikasi Monitor Lelang** adalah aplikasi online yang diakses oleh peserta lelang untuk melihat rekaman setoran dan juga harga hasil lelang serta untuk melihat berapa uang hasil lelang dan status sudah diambil atau belum.

Metode pengembangan *waterfall* digunakan dalam mengembangkan aplikasi. Metode *waterfall* yang dibangun oleh *Winston W. Royce* pada tahun 1970 untuk mengatur praktik produk perangkat lunak. 5 Fase pada model *waterfall* yaitu *analysis, design, implementation, testing* dan *maintenance*.

3. HASIL DAN PEMBAHASAN

Berdasarkan penelitian yang telah dilakukan, maka dilakukan pengujian aplikasi akhir berbasis local host dimana server *local* tetap bisa terhubung ke internet untuk bisa mengambil data dan menguji juga saat koneksi internet tidak ada ternyata aplikasi local host tetap bisa berjalan normal dalam menampilkan informasi.

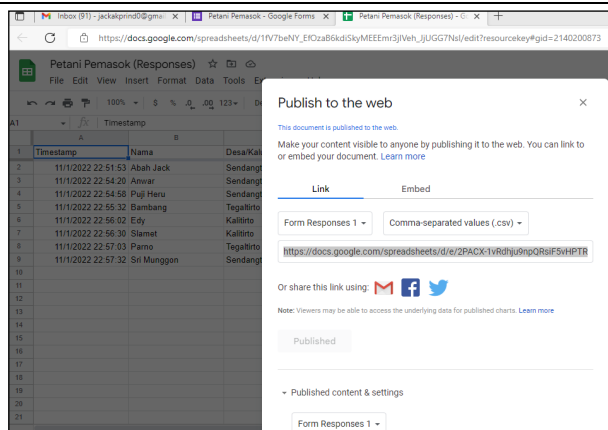
3.1 Data Share Google Sheet

Dalam penelitian ini data utama disediakan oleh *Google Sheet* dimana data tersebut adalah hasil transaksi aplikasi lain, Gambar 2 menunjukkan *Share Google Sheet* dari sebuah sheet. Dimana *Sheet* tersebut di share public dengan format CSV (*Comma-Separated Values*). Data ini yang akan diambil oleh aplikasi untuk kemudian di save pada cache local. Adapun tautan dari share tersebut adalah sebuah URL sebagai berikut:

https://docs.google.com/spreadsheets/d/e/2PACX-1vRdhju9npQRsiF5vHPTRgWQjcvCTM_nE3YHtoj90borcV7gSEvvK7mKvc0Vr6nOZXY19ileLqS1TG2N/pub?gid=2140200873&single=true&output=csv

Adapun data hasil dari tautan diatas akan berbentuk sebuah file dengan format CSV sebagai berikut:

```
Timestamp, Nama, Desa/Kalurahan, Alamat, Nomor Telpn, Produk
11/1/2022 22:51:53, Abah Jack, Sendangtirto, Klodangan 01/26, 081327780105, Cabai
11/1/2022 22:54:20, Anwar, Sendangtirto, Klodangan , 0192838372732, "Jagung, Sayuran Daun"
11/1/2022 22:54:58, Puji Heru, Sendangtirto, Klakah, 09182828273, Mentimun
11/1/2022 22:55:32, Bambang, Tegaltirto, Tegal, 09829272821, Cabai
11/1/2022 22:56:02, Edy, Kalitirto, Kali, 0928292821, Cabai
11/1/2022 22:56:30, Slamet, Kalitirto, Kali, 08128373726, "Cabai, Mentimun"
11/1/2022 22:57:03, Parno, Tegaltirto, Kuton, 0817263735272, "Cabai, Tremes"
11/1/2022 22:57:32, Sri Munggon, Sendangtirto, Munggon, 082847463736, "Sayuran Daun, Tremes"
```

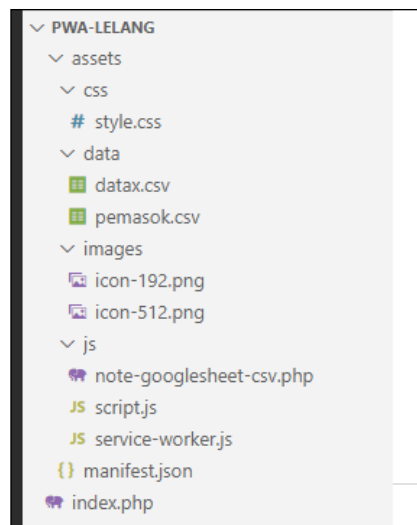


Gambar 2. Share Google Sheet

3.2. Progressive Web Application (PWA)

Dalam penelitian ini, pemanfaatan PWA menjadi bagian utama untuk menampilkan output informasi. Adapun struktur PWA bisa dilihat pada gambar 3, dimana sebuah web memiliki folder *assets* yang terdiri dari sub-folder *css, data, images* dan *js*. Dimana sub-folder *data* adalah sub-folder yang digunakan untuk menampung/cache data dari server pusat dalam hal ini hasil dari *share google sheet* sehingga aplikasi saat

mengambil data akan mengambil data dari sub-folder ini dan hanya akan mengambil data dari *share google sheet* pada saat-saat tertentu saja.



Gambar 3. Struktur Folder Aplikasi berbasis PWA

3.2.1 Manifest

Agar pengguna bisa mendapatkan akses cepat dan menikmati layar penuh tampilan seperti dengan aplikasi asli maka dibutuhkan sebuah *manifest* yang diletakkan pada sub-folder *assets* sebagai berikut

```
{
  "short name": "PWALelang",
  "name": "PWA Pasar Lelang",
  "icons": [
    {
      "src": "/pwa-lelang/assets/images/icon-192.png",
      "type": "image/png",
      "sizes": "192x192"
    },
    {
      "src": "/pwa-lelang/assets/images/icon-512.png",
      "type": "image/png",
      "sizes": "512x512"
    }
  ],
  "start url": "/pwa-lelang/",
  "background color": "#ffffff",
  "display": "standalone",
  "scope": "/pwa-lelang/",
  "theme color": "#ffffff"
}
```

3.2.2 Service Worker

Digunakan untuk menentukan tempat/lokasi *cache*, sehingga pemetaan data-data yang harus disimpan pada *cache* akan tertata dan mudah untuk di Kelola. Termasuk data *cache* yang diperoleh dari server/URL.

```
const nameCache = "pwa-penelitian";
const allCache = [
  "/pwa-lelang/",
  "/pwa-lelang/assets/css/style.css",
  "/pwa-lelang/assets/js/script.js",
  "/pwa-lelang/assets/data/pemasok.csv",
];
self.addEventListener("install", (event) => {
```

```
event.waitUntil( async() => {
  const cache = await caches.open(nameCache);
  await cache.addAll(allCache);
});
});
self.addEventListener("fetch", (event) => {
  event.respondWith(
    caches.open(nameCache).then((cache) => {
      cache.match(event.request).then((cacheResponse) => {
        fetch(event.request).then((networkResponse) => {
          cache.put(event.request, networkResponse);
        });
        return cacheResponse || networkResponse;
      });
    });
});
```

3.3. File *index.html*

File *index.html* sebagai file pertama yang akan diload saat aplikasi dijalankan, dimana file ini akan memanager semua resource yang ada. Berikut adalah script file *index.html*

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Pasar Lelang Hasil Pertanian with PWA</title>
  <link rel="manifest" href="/pwa-lelang/assets/manifest.json" />
  <link href="/pwa-lelang/assets/css/style.css" rel="stylesheet" />
  <link rel="icon" href="/pwa-lelang/assets/images/icon-192.png" type="image/png"
/>
  <link rel="apple-touch-icon" href="/pwa-lelang/assets/images/icon-192.png">
  <meta name="theme-color" content="white"/>
  <meta name="apple-mobile-web-app-capable" content="yes">
  <meta name="apple-mobile-web-app-status-bar-style" content="black">
  <meta name="apple-mobile-web-app-title" content="PWA-Lelang">
  <meta name="msapplication-TileImage"
content="/pwa-lelang/assets/images/icon-192.png">
  <meta name="msapplication-TileColor" content="#FFFFFF">
</head>
<body>
<h1>Pasar Lelang Hasil Pertanian</h1>
<select id="pelapor">
  <option value="">Semua Pemasok</option>
</select>
<table border="1">
  <thead><th>Nama Pemasok</th><th>Desa</th><th>Alamat</th><th>Telpon</th>
  <th>Hasil Pertanian</th></thead>
  <tbody id="data"></tbody>
</table>
<script src="/pwa-lelang/assets/js/script.js"></script>
<script>
var selectPelapor = document.getElementById("pelapor");
var tableData = document.getElementById("data");
window.addEventListener("load", () => {
  data(selectPelapor.value);
  selectData();});
async function selectData() {
  const resData = await dataPelapor();
  resData.forEach(data => {
    selectPelapor.options.add(new Option(data, data));
  });}
async function data(selectData) {
  const resData = await loadData(selectData);
  var result = '';
  resData.forEach(data => {
    result += `
      <tr><td>${data[1]}</td><td>${data[2]}</td><td>${data[3]}</td>
      <td>${data[4]}</td><td>${data[5]}</td></tr>`;
  });
```



```
tableData.innerHTML = result;}
selectPelapor.addEventListener('change', (e) => {
  const selectData = selectPelapor.value ? selectPelapor.value : '';
  data(selectData) })
</script>
</body>
```

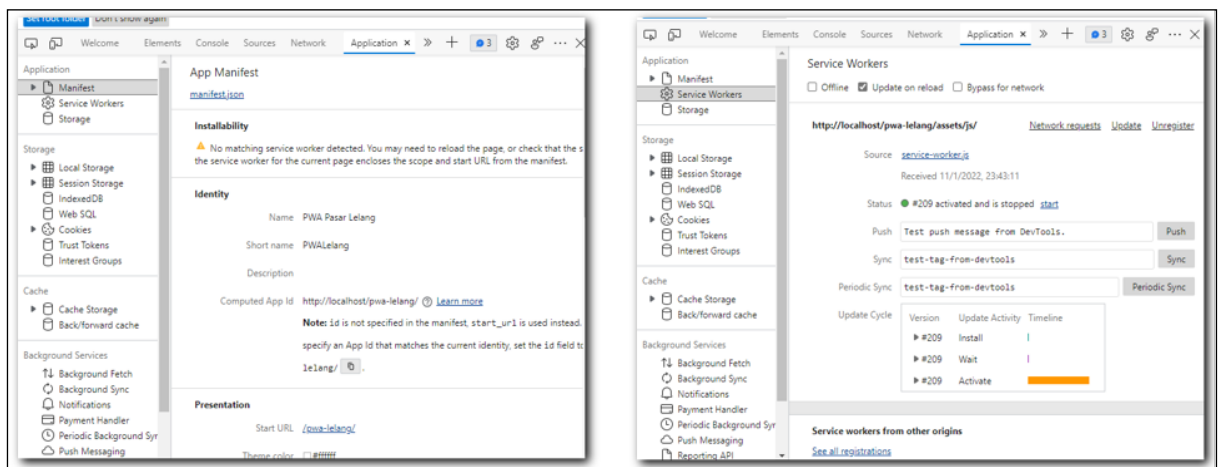
3.4. Output Aplikasi

Setelah semua proses coding dilakukan dan aplikasi dijalankan, diperoleh aplikasi berbasis PWA terlihat pada gambar 4 sebuah tampilan aplikasi.



Nama Pemasok	Desa	Alamat	Telpon	Hasil Pertanian
Abah Jack	Sendangtirto	Klodangan 01/26	081327780105	Cabai
Anwar	Sendangtirto	Klodangan	0192838372732	"Jagung
Puji Heru	Sendangtirto	Klakah	09182828273	Mentimun
Bambang	Tegaltirto	Tegal	09829272821	Cabai
Edy	Kalitirto	Kali	0928292821	Cabai
Slamet	Kalitirto	Kali	08128373726	"Cabai

Gambar 4. Aplikasi PWA Lelang



Gambar 5. Manifest dan Service-Worker

Pada saat aplikasi berjalan, maka dengan fasilitas *inspect* pada browser bisa dimonitor tentang *manifest* dan *service-worker* seperti pada gambar 5.

4. KESIMPULAN

Dari hasil penelitian ini dapat disimpulkan bahwa penerapan PWA sekumpulan data bisa diakses dari berbagai aplikasi dan data tersebut bisa terdistribusikan keberbagai aplikasi menggunakan *cache* yang telah ditentukan, sehingga jika *cache* telah terisi, maka saat tidak ada koneksi dengan penyedia data utama, maka aplikasi berbasis PWA tetap akan bisa berjalan dengan sempurna.

UCAPAN TERIMA KASIH

Ucapan terimakasih kami haturkan kepada:

1. Intitut Sains dan Teknologi AKPRIND Yogyakarta yang telah membiayai penelitian.
2. LPPM Institut Sains dan Teknologi AKPRIND Yogyakarta yang telah meloloskan penelitian.
3. Jurusan Informatika yang telah memberikan fasilitas laboratorium untuk melakukan penelitian.
4. Dan para pihak yang tidak bisa kami sebutkan, yang telah mendorong dan memberi semangat dalam penelitian

DAFTAR PUSTAKA

- Groff, J., Weinberg, P., & Opperl, A. (2010). *SQL The Complete Reference - Third Edition*. United States: The McGraw-Hill Companies.
- Marrs, T. (2017). *JSON at Work*. . Sebastopol: O'Reilly Media inc.
- Smith, B. (2015). *Beginning JSON*. New York: Apress.
- Szklarski, M. (2019), *PWA in Practice*. Poland: e-Point
- Haryani, P. (2016). Evaluasi Kualitas Layanan E-Government Pemerintah Kota Yogyakarta Dengan Metode E-GovQual Modifikasi. *Simposium Nasional Ke-15 RAPI 2016* (pp. 379-386). Surakarta: Fakultas Teknik Universitas Muhammadiyah Surakarta.
- Triyono, J. (2015). Sistem Informasi Agroteknologi berbasis Web dan Jejaring Sosial Twitter. *Seminar Nasional IENACO* (pp. 205-212). Surakarta: UMS Surakarta.
- Triyono, J., & Haryani, P. (2017). Membangun Aplikasi Front End Web Dosen Berbasis Web APPML (Application Modeling Language). *Simposium Nasional RAPI XVI – 2017 FT UMS* (pp. 289-298). Surakarta: FT UMS.
- Triyono, J., Haryani, P., & Farid AS, M. (2018). Membangun Aplikasi Web Front End dengan Teknik JSON (Java Script Object Notation). *Seminar Nasional Aplikai Sains dan Teknologi (SNAST 2018)* (pp. B237-B246). Yogyakarta: IST AKPRIND Yogyakarta.
- Triyono, J., Fatkhiyah, E., Wicaksana, E, M, (2021). Perancangan Arsitektur Aplikasi Tersebar Berbasis File Text. *Seminar Multimedia & Artificial Intellegence (SMAI-2021)*, Yogyakarta: Universitas Mercu Buana Yogyakarta
- Lorenz, (2022), *How to work with JSON in JavaScript and PHP*. Diakses tanggal 11 November 2022, dari <https://webdeasy.de/en/using-json-in-javascript-and-php/>