

Rancang Bangun Aplikasi E-commerce Berbasis Progressive Web Apps Pada Distro Romusha

by Devi Anggraeni .

FILE	JURNAL_TA.DOC (1.31M)	WORD COUNT	2018
TIME SUBMITTED	31-JUL-2018 12:21PM (UTC+0700)	CHARACTER COUNT	12578
SUBMISSION ID	986501908		

Rancang Bangun Aplikasi *E-commerce* Berbasis *Progressive Web Apps* Pada Distro Romusha

Devi Anggraeni

Program Studi Teknik Informatika, Universitas 17 Agustus 1945 Surabaya

Jalan Semolowaru no. 45, Surabaya, 60118

deviianggraeni@gmail.com

Abstract

Progressive Web Apps (PWA) is one of the newest technologies used to combine web and mobile apps. This technology in its use will feel like an application. However it will eliminate the download and install stages in the app store. Where the process can reduce the interest of app users to use it. This application will be created using web technology in general but in implementing it using PWA Builder tools. Stages of developing the web to PWA is to generate the first manifest then build service worker after that publish PWA. Research with progressive web apps technology is applied to Romusha Distro business activities.

Romusha Distribution is one of the business companies engaged in the sale of clothing and shoes. In the current era of technology Romusha Distro requires service and improvement improvements in its business processes. By applying progressive web apps technology is expected to increase customer satisfaction and have advantages compared with other businesses.

Keywords: *Progressive Web Apps, PWABuilder, e-commerce*

Abstrak

Progressive Web Apps (PWA) merupakan salah satu teknologi terbaru yang digunakan untuk menggabungkan aplikasi web dan seluler. Teknologi ini dalam penggunaannya akan terasa seperti sebuah aplikasi. Namun akan menghilangkan tahap download dan install di toko aplikasi. Dimana proses tersebut dapat mengurangi minat pengguna aplikasi untuk menggunakannya. Aplikasi ini akan dibuat dengan menggunakan teknologi web pada umumnya namun dalam pengimplementasiannya menggunakan tools PWA Builder. Tahapan mengembangkan web ke PWA yaitu membuat generate manifest dahulu kemudian build service worker setelah itu publish PWA. Penelitian dengan teknologi progressive web apps ini diterapkan pada kegiatan bisnis Distro Romusha.

Distro Romusha merupakan salah satu perusahaan bisnis yang bergerak dibidang penjualan pakaian dan sepatu. Di era teknologi saat ini Distro Romusha membutuhkan peningkatan layanan dan perbaikan dalam proses bisnisnya. Dengan menerapkan teknologi progressive web apps ini diharapkan akan meningkatkan kepuasan konsumen dan mempunyai keunggulan dibandingkan dengan bisnis lain.

Kata kunci: *Progressive Web Apps, PWABuilder, e-commerce*

1. PENDAHULUAN

² Pada dasarnya *E-Commerce* merupakan dampak dari berkembangnya teknologi informasi dan telekomunikasi, sehingga secara signifikan merubah cara manusia melakukan interaksi dengan lingkungannya, yang dalam hal ini adalah terkait dengan mekanisme dagang. *E-Commerce* bukanlah sekedar mekanisme penjualan barang atau jasa melalui medium internet, tetapi lebih pada sebuah transformasi bisnis yang merubah cara-cara perusahaan dalam melakukan aktivitas usahanya sehari-hari [1].

Progressive web apps memanfaatkan teknologi terbaru untuk menggabungkan aplikasi web dan aplikasi seluler. Anggap saja sebagai situs web yang dibuat menggunakan teknologi web namun bertindak dan terasa seperti sebuah aplikasi. Kemajuan terbaru di browser dan ketersediaan *API Cache* memungkinkan pengembang web mengizinkan pengguna memasang aplikasi web ke layar awal mereka, menerima pemberitahuan dan bahkan bekerja secara *offline* [2].

Pada aplikasi *mobile* umumnya seorang pengguna pertama-tama harus mencari aplikasi di toko aplikasi, mendownloadnya, menginstalnya dan kemudian buka. Dan saat pengguna kembali ke aplikasi, mereka akan diminta memasang aplikasi dan meningkatkan versi. Ketika pengguna menemukan *aplikasi progressive web apps*, mereka akan dapat segera mulai menggunakannya, menghilangkan tahap pengunduhan dan penginstalan yang tidak perlu [3]. Selain itu, aplikasi seluler yang dirancang dengan baik menghabiskan lebih sedikit data dan lebih cepat karena beberapa sumber berada pada perangkat.

2. METODE PENELITIAN⁶

Agar hasil yang dicapai tidak menyimpang dari tujuan yang telah ditentukan sebelumnya penulis menggunakan metodologi penelitian sebagai pedoman melaksanakan penelitian. Berikut Metode Penelitian Lapangan (*Field Research*) yang digunakan :

a. Pengambilan Data

Pengambilan data dilakukan di Distro Romusha yang berada di⁶ Jalan Pahlawan No 1 Sidoarjo, dimana penulis melakukan pengumpulan data dengan cara bertanya langsung kepada pihak terkait.

b. Pengolahan Data

Data-data yang diperoleh kemudian diolah untuk dilakukan analisa.

⁴ Secara umum, dapat dikatakan bahwa alur sistem ini berisi "siapa saja yang memberi data (data apa saja) ke sistem, serta kepada siapa saja informasi (informasi apa saja) yang harus dihasilkan". Jadi data yang dibutuhkan adalah sebagai berikut :

1. Siapa saja pihak yang akan memberikan data.
2. Data apa saja yang diberikan.
3. Kepada siapa sistem harus memberi informasi.
- ⁵ 4. Apa saja isi informasi yang harus dihasilkan.

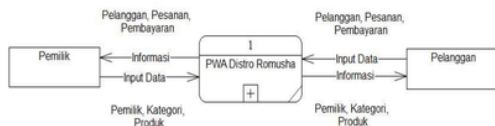
Selanjutnya data yang diberikan pengguna kepada sistem adalah sebagai berikut :

1. Profil pengguna
2. Item yang dibeli.
3. Uang pembayaran.

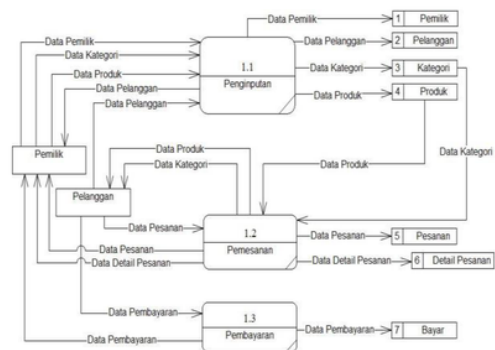
Informasi yang diberikan sistem kepada pengguna adalah sebagai berikut :

1. Item yang dijual.
2. Jumlah uang yang harus dibayar pengguna.

Dari informasi yang dibutuhkan dan diberikan sistem tersebut, maka DFD Context dan DFD Level 1 sebagai berikut :



Gambar 1. DFD Context PWA Distro Romusha



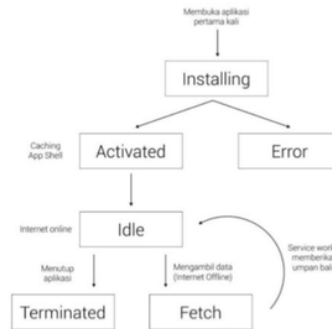
Gambar 2. DFD Level 1 Penjualan Distro Romusha

a. Service Worker

Service Worker merupakan salah satu jenis dari web worker yang berjalan di belakang browser pengguna. Service worker pada dasarnya adalah berkas JavaScript yang berjalan pada thread yang berbeda dengan main thread browser, menangani network request, caching, mengembalikan resource dari cache [4]. Beberapa aset web dapat disimpan sebagai cache local pada service worker, sehingga dengan jaringan internet yang kurang memadai pun, pengguna tetap mendapat pengalaman yang baik saat menggunakan aplikasi.

Untuk memasang service worker ke laman aplikasi, kita butuh mendaftarkannya terlebih dahulu dengan menggunakan

JavaScript yang ada di halaman web. Setelah diregistrasi, tahap install service worker di latar akan dimulai oleh browser. Pada service worker, strategi caching dapat diaplikasikan sesuai keinginan kita, namun tidak ada satu strategi terbaik untuk caching konten dinamis, dan ada banyak situasi yang bisa mempengaruhi strategi caching [4].



Gambar 3. Daur hidup (life cycle). Service worker

8

b. Web App Manifest

Web App Manifest merupakan file JSON sederhana yang memberikan developer kemampuan untuk mengontrol bagaimana aplikasi terlihat oleh pengguna (misalnya, layar beranda perangkat seluler), mengarahkan apa yang bisa diluncurkan pengguna, dan menentukan tampilannya pada saat peluncuran. Web App Manifest menyediakan kemampuan untuk menyimpan bookmark situs ke layar beranda perangkat. Ketika sebuah situs diluncurkan dengan cara ini:

- Situs akan memiliki ikon dan nama yang bisa membedakannya dari situs yang lain.
- Situs akan menampilkan sesuatu kepada pengguna selagi sumber daya diunduh atau dipulihkan dari cache.

Dengan memasang web app manifest bisa menetapkan set ikon untuk digunakan browser, memuat laman web dari

jaringan dan memerlukan waktu lebih dari satu atau dua detik, menampilkan materi di laman beranda dan untuk menyediakan pengalaman pengguna yang lebih baik bisa mengganti layar putih dengan judul, warna, dan gambar.

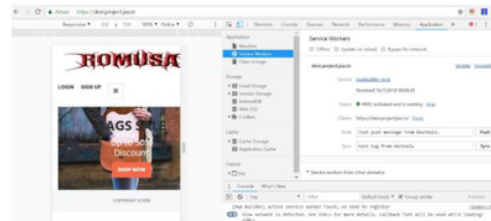
c. Lighthouse

Lighthouse merupakan alat (bantu) terbuka untuk meningkatkan kualitas aplikasi web Anda. Anda bisa menjalankannya sebagai Ekstensi Chrome. Cara mengoprasikannya yaitu dengan memberi *Lighthouse* sebuah URL yang ingin Anda audit, maka *Lighthouse* menjalankan serangkaian pengujian terhadap halaman tersebut, kemudian menghasilkan sebuah laporan mengenai seberapa optimal laman tersebut. Dari sini pengujian yang dinyatakan tidak lulus bisa digunakan sebagai indikator atas apa saja yang bisa Anda lakukan untuk meningkatkan aplikasi.

1
Dari hasil yang dihasilkan oleh *Lighthouse* kita bisa melihat posisi *webapps* kita sudah optimal atau belum, jika belum tinggal mencari bagian mana yang belum optimal untuk di optimalkan. Jika salah satu point belum optimal, maka nanti akan muncul tanda x dengan warna merah.

3. HASIL DAN PEMBAHASAN

Penambahan *service worker* jika berhasil akan terlihat status *activated* dan *running*. Setelah terpasang *service worker* pada aplikasi meskipun tiba-tiba *offline* karena jaringan tidak stabil, laman seharusnya masih bisa dimuat ulang sepenuhnya. Alasan aplikasi ini dapat berfungsi secara *offline* adalah dukungan *offline* dari *service worker* yang telah terpasang.

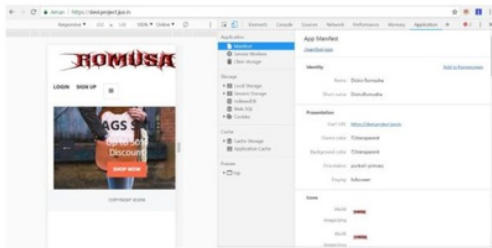


Gambar 4. Tampilan *Service Worker* Terpasang

Jika penambahan *app manifest* berhasil akan terlihat pada *point manifest*. Untuk membuktikan bahwa pengguna dapat menambahkan situs ke layar beranda (*homescreen*), pengguna bisa membuka URL aplikasi di google dan akan muncul notifikasi apakah pengguna akan menambahkan set ikon untuk digunakan browser pada *homescreen*.

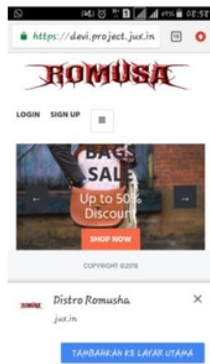
Chrome DevTools menyediakan versi properti manifest yang mudah dibaca. Lihat *Manifest* aplikasi web pada dokumen *Chrome DevTools* untuk informasi selengkapnya mengenai ini. Jika ingin pendekatan otomatis terhadap validasi *manifest* aplikasi web, lihat *Lighthouse*. *Lighthouse* adalah alat (bantu) pengauditan aplikasi web yang dijalankan sebagai Ekstensi Chrome atau sebagai modul NPM. Hanya perlu menyediakan URL pada *Lighthouse*, kemudian *Lighthouse* akan menjalankan paket audit terhadap laman tersebut, dan selanjutnya menampilkan hasilnya di laporan. Audit *Lighthouse* yang terkait dengan manifes aplikasi web antara lain akan memeriksa apakah:

- Aplikasi bisa ditambahkan ke layar beranda.
- Setelah ditambahkan, aplikasi akan diluncurkan bersama layar pembuka khusus.
- Warna bilah alamat browser akan disesuaikan.
- Aplikasi berada di HTTPS (prasyarat untuk Add to *Homescreen*).



Gambar 5. Tampilan *App Manifest* Terpasang

Untuk membuktikan bahwa pengguna dapat menambahkan situs ke layar beranda (*homescreen*), pengguna bisa membuka URL aplikasi di google dan akan muncul notifikasi apakah akan menambahkan set ikon untuk digunakan *browser* pada *homescreen*.



Gambar 6. Tampilan Permintaan *Add Homescreen* Pada *Mobile*

Setelah muncul notifikasi tersebut pilih untuk tambahkan ke layar utama (*homescreen*). Tunggu proses penambahan karena dibutuhkan waktu beberapa detik untuk memasang. Setelah terpasang akan ada notifikasi kembali bahwa jalan pintas sudah terpasang. Proses ini merupakan dukungan dari *app manifest* yang telah terpasang pada aplikasi.

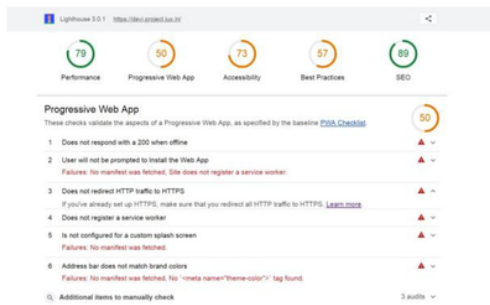


Gambar 7. Tampilan *Homescreen* Aplikasi Berhasil Pada *Mobile*

Pengujian dilakukan untuk mengukur tingkat keberhasilan dan keefisienan aplikasi yang telah dibangun. Metode uji yang diterapkan yaitu mengukur nilai *Progressive Web Apps* pada aplikasi menggunakan *lighthouse* yang sudah tersedia di dalam *chrome*.

Untuk menilai seberapa baik PWA pada aplikasi, Google sudah menyiapkan *tool Lighthouse*. *Tool* ini selain tersedia dalam bentuk *Chrome Extension*, juga bisa digunakan sebagai *npm package*. *Lighthouse* ini akan mengukur seberapa lengkap komponen PWA yang kita implementasikan. Semakin lengkap fitur PWA yang kita buat, dan semakin baik implementasinya (menggunakan *best practice* yang direkomendasikan) akan semakin tinggi skor PWA-nya.

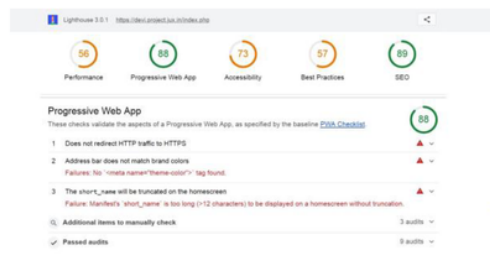
1 Dari hasil yang dihasilkan oleh *Lighthouse* kita bisa melihat posisi web apps kita sudah optimal atau belum, jika belum tinggal mencari bagian mana yang belum optimal untuk di optimalkan. Jika salah satu poin belum optimal, maka nanti akan muncul tanda dengan warna merah. Berikut merupakan gambar hasil pengujian dengan *lighthouse* sebelum menjadi PWA:



Gambar 8. Hasil Pengujian dengan *lighthouse* sebelum PWA

Pada hasil pengujian dengan *lighthouse* sebelum PWA terlihat hasil untuk *Progressive web apps* pada alamat URL tersebut hanya mencapai angka 50. Masih banyak poin yang belum ada pada aplikasi untuk di katakan sebagai aplikasi *Progressive webs apps*. Poin yang bertanda merah sebagian besar merupakan poin-poin yang didukung dengan *service worker* dan *app manifest*. Maka untuk bisa dikatakan sebagai aplikasi *progressive web apps* perlu penambahan *service worker* dan *app manifest* pada aplikasi.

Setelah dilakukan penambahan *service worker* dan *app manifest* pada aplikasi, berikut gambar dari hasil pengujian dengan *lighthouse* :



Gambar 9. Hasil Pengujian dengan *lighthouse* sesudah PWA

Pada hasil pengujian dengan *lighthouse* setelah penambahan *service worker* dan *app manifest*, aplikasi sudah bisa dikatakan sebagai aplikasi berbasis *progressive web apps* dengan mencapai

angka 88. Aplikasi belum sepenuhnya memenuhi poin audit pada *lighthouse*. Terdapat 3 poin yang masih berwarna merah dan perlu diadakan perbaikan untuk pengembangan aplikasi tersebut. Namun pengguna aplikasi sudah bisa merasakan pemasangan pada homescreen tanpa harus mendownloadnya melalui toko aplikasi. Selain itu pengguna juga bisa merasakan pengalaman *offline* dan tidak akan memunculkan pesan *network error* atau *white screen* seperti yang selama ini seringkali kita alami ketika berada pada jaringan yang sangat buruk.

4. SIMPULAN

Setelah melakukan pengolahan dan analisis data pada penelitian ini, didapatkan beberapa kesimpulan sebagai berikut:

1. Berdasarkan hasil penelitian penulis dapat menyimpulkan bahwa aplikasi *e-commerce* pada Distro Romusha berbasis *progressive web apps* ini dapat menerapkan prinsip-prinsip dari metode *progressive web apps* itu sendiri seperti, dapat berjalan saat *offline*, tidak akan memunculkan pesan *network error* atau *white screen* seperti yang selama ini seringkali kita alami ketika berada pada jaringan yang sangat buruk dan membuat *manifest* aplikasi web sehingga menghilangkan proses *download* aplikasi melalui toko aplikasi dan langsung bisa memasang pada *homescreen* secara langsung.
2. Dengan adanya aplikasi *e-commerce* berbasis *progressive web apps* ini pada Distro Romusha dapat membantu dalam menyampaikan informasi produk kepada konsumen.

DAFTAR PUSTAKA

- [1] D. Wirdasari, "Teknologi E-commerce Dalam Proses Bisnis," *Jurnal SAINTIKOM*, vol. 7, pp. 324-335, 2009.
- [2] R. S. Michla, "Progressive WEBAPP : Review," *International Research Journal of Engineering and Technology (IRJET)*, vol. 03, no. 06, p. 5, 2016.
- [3] D. V. Karpagam, L. R, P. R and Priyadharshini.S, "PERFORMANCE ENHANCEMENT OF WEBPAGE USING PROGRESSIVE WEB APP FEATURES," *International Journal of Innovative Research in Advanced Engineering (IJIRAE)*, vol. 4, no. 3, pp. 1-7, 2017.
- [4] J. A. W. N. K. Laurensius Adi, "Platform E-Learning untuk Pembelajaran Pemrograman Web Menggunakan Konsep Progressive Web Apps," *JURNAL TEKNIK ITS Vol. 6, No. 2 (2017), 2337-3520 (2301-928X Print)*, vol. 6, pp. A579 - A583, 2017.

Rancang Bangun Aplikasi E-commerce Berbasis Progressive Web Apps Pada Distro Romusha

ORIGINALITY REPORT

% **19**
SIMILARITY INDEX

% **18**
INTERNET SOURCES

% **2**
PUBLICATIONS

% **9**
STUDENT PAPERS

PRIMARY SOURCES

1 webappid.blogspot.com %**4**
Internet Source

2 www.slideshare.net %**3**
Internet Source

3 eprasetianti.blogspot.co.id %**2**
Internet Source

4 achmadsolikun.blogspot.com %**2**
Internet Source

5 bhegrezfack.blogspot.com %**2**
Internet Source

6 repository.usu.ac.id %**1**
Internet Source

7 Submitted to Central Queensland University %**1**
Student Paper

8 Submitted to Universitas Brawijaya %**1**
Student Paper

9 ejurnal.its.ac.id

Internet Source

% 1

10

Submitted to Universitas Muria Kudus

Student Paper

% 1

11

docplayer.info

Internet Source

<% 1

12

publikasiilmiah.ums.ac.id

Internet Source

<% 1

13

www.kehamilanku.web.id

Internet Source

<% 1

EXCLUDE QUOTES OFF

EXCLUDE MATCHES OFF

EXCLUDE
BIBLIOGRAPHY OFF