

Perbandingan Performa Progressive Web Apps dan Mobile Web Terkait Waktu Respon, Penggunaan Memori dan Penggunaan Media Penyimpanan

Muhammad Rasyid Ridho¹, Aryo Pinandito², Ratih Kartika Dewi³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹muh.r.ridho@gmail.com, ²aryo@ub.ac.id, ³ratihkartikad@ub.ac.id

Abstrak

Progressive Web App (PWA) adalah Mobile Web yang menggunakan teknologi-teknologi terbaru guna menghasilkan Mobile Web yang memiliki pengalaman pengguna yang jauh lebih baik dari Mobile Web tradisional. Untuk menghasilkan aplikasi *web* dengan pengalaman pengguna yang baik, tentu tidak lepas dari performa dari aplikasi dan teknologi dibaliknya. Berbicara tentang performa, banyak hal yang dapat diperhitungkan, beberapa diantaranya adalah waktu respon, penggunaan memori dan penggunaan media penyimpanan. Pada penelitian ini akan membahas mengenai perbandingan antara PWA dan Mobile Web terkait hal-hal tersebut agar dapat memberikan bahan pertimbangan dan referensi untuk penerapan teknologi yang baru dikenalkan ini. Dari penelitian ini didapatkan beberapa kesimpulan, yaitu kecepatan dalam memuat data menyesuaikan dengan ukuran berkas dan *cache* yang digunakan serta frekuensi pengaksesan 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. Untuk performa terkait penggunaan memori, Mobile Web menggunakan memori yang lebih sedikit dibandingkan dengan PWA dikarenakan adanya proses tambahan pada PWA (Service Worker). Sedangkan untuk performa terkait penggunaan media penyimpanan, pada Mobile Web tidak menggunakan media penyimpanan sama sekali, sedangkan pada PWA penggunaan media penyimpanan menyesuaikan dengan *cache* yang disimpan pada peramban.

Kata kunci: *web, mobile, progressive, aplikasi, performa*

Abstract

Progressive Web Apps (PWA) is a Mobile Web that uses the latest technologies to create a Mobile Web with better user experiences than traditional Mobile Web. Creating a web application with such experiences, one thing that can not be forgotten is performances and the technologies behind it. Speaking of performance, many aspects could be considered, such as responses time, memory usages and storage usages. Thus, this study will discuss about comparisons between PWA and Mobile Web related to those aspects in order to provide some insights and references of this newly introduced technology. From this study, there are some things that can concluded. Response time performance depends on page resources size, caches size and the access frequencies. On a small sized page and small sized cache files, Mobile Web still faster than Progressive Web App, while on large sized page and cache files Progressive Web App is the winner. In memory usages performance, Mobile web uses memory fewer than PWA due to an additional process that PWA has (Service Worker). In storage usages performance, Mobile web is not using any of storage spaces, while the storage usages of PWA depends on the size of the caches stored in the browser.

Keywords: *web, mobile, progressive, application, performance*

1. PENDAHULUAN

Dewasa ini, jumlah pengguna internet di dunia terus berkembang. Berkembangnya pengguna internet ini tidak lepas dari semakin mudahnya orang-orang dalam mengakses

internet. Salah satu hal yang memudahkan orang-orang dalam mengakses internet adalah makin banyaknya perangkat yang mampu terhubung dengan internet, salah satunya adalah perangkat bergerak.

Selain peningkatan penetrasi pengguna, varian teknologi yang mendukung penggunaan

internet di perangkat bergerak pun bermunculan. Salah satunya adalah teknologi dalam hal performa dan pengalaman pengguna yaitu Progressive Web Apps (PWA). PWA tidak dikemas dan dipublikasikan lewat App Store, PWA hanyalah Mobile Web yang diberi kelebihan-kelebihan yang mampu membuat Mobile Web menjadi *top-level* pada *task switcher*, *homescreen* dan *notification tray* tanpa harus instalasi di awal (Russell, 2015). Dari sisi implementasi, PWA menggunakan beberapa pendekatan, mulai dari pendekatan dari pengoptimisasian aplikasi, penggunaan teknologi-teknologi anyar, hingga standarisasi Mobile Web. Hal ini diklaim mampu meningkatkan performa guna meningkatkan kualitas pengalaman pengguna.

Berbicara tentang performa aplikasi tentu banyak hal yang harus diperhitungkan. Salah satunya adalah waktu respon dari aplikasi tersebut. Permasalahan waktu respon yang lambat adalah permasalahan klasik yang dihadapi oleh pengembang aplikasi-aplikasi di dunia. Hal ini dibuktikan oleh beberapa data dan penelitian sebelumnya, seperti data yang dilansir oleh Akamai Technology (2016) menunjukkan rata-rata kecepatan respon aplikasi *web* yang masih berkisar di angka 5000 mdtk hingga 6000 mdtk pada aplikasi di perangkat bergerak (Belson, 2016).

Mengatasi permasalahan ini PWA menawarkan beberapa solusi, diantaranya adalah Service Worker, Application Shell, Web App Manifest, dll. Ada hal yang perlu diperhatikan dari solusi-solusi tersebut, yaitu penggunaan memori, contoh saja cara kerja dari Service Worker yang terpisah dari halaman *web* dikarenakan Service Worker adalah proses yang berjalan di balik layar sehingga perlu diketahui seberapa besar penggunaan memori dalam PWA dibandingkan dengan Mobile Web tradisional. Mengingat penggunaan memori adalah salah satu faktor penentu baik buruknya performa sebuah aplikasi pada perangkat bergerak (AppDynamics, 2014).

Selain penggunaan memori, ada hal lain yang perlu diperhatikan yaitu penggunaan media penyimpanan. Karena, pada PWA beberapa data-data aplikasi (contoh: *cache*) akan disimpan pada *browser*. Padahal, tiap perangkat memiliki batasan penyimpanan (Osmani, 2016). Oleh karena itu dirasa perlu untuk mengetahui seberapa besar kebutuhan media penyimpanan oleh PWA dibandingkan dengan Mobile Web tradisional.

Berdasarkan latar belakang tersebut, perlu dilakukan penelitian untuk mengetahui hasil dari perbandingan performansi PWA dan Mobile Web agar dapat memberikan bahan pertimbangan untuk penerapan teknologi PWA agar mengurangi permasalahan-permasalahan yang dialami pengembang dan pengguna aplikasi pada perangkat bergerak serta dapat dijadikan sebagai bahan referensi untuk para peneliti dalam mengembangkan dan mendalami PWA pada perangkat bergerak.

2. METODOLOGI

Metodologi adalah tahapan-tahapan yang digunakan dalam menyelesaikan penelitian ini. Adapun tahapannya terdiri dari studi literatur gai langkah untuk mengumpulkan dan mempelajari literatur dari beberapa bidang ilmu yang berkaitan dalam penelitian ini. Analisis kebutuhan sebagai persiapan dalam menyiapkan bahan-bahan yang akan dibutuhkan untuk melakukan penelitian ini. Rancangan objek kajian yang membahas bagaimana bentuk antarmuka, struktur berkas, alur pertukaran data dan manajemen *cache* pada aplikasi. Pengujian yang merupakan metode yang dilakukan peneliti untuk mendapatkan data-data yang akan digunakan untuk melihat performa dari aplikasi yang diuji. analisis hasil pengujian yang membahas mengenai hasil dan analisis dari hasil pengujian pada penelitian ini. Kesimpulan dan saran yang berisi kesimpulan dan saran yang didapat dari hasil penelitian ini.

3. PEMBAHASAN

3.1. Analisis Kebutuhan

a. Perangkat Lunak

Pada penelitian ini membutuhkan total empat perangkat lunak pada perangkat komputer dan perangkat bergerak. Pada perangkat komputer membutuhkan 3 perangkat lunak, yaitu XAMPP sebagai peladen, Google Chrome sebagai aplikasi untuk melihat data-data hasil pengujian dan ADB Driver untuk menghubungkan Google Chrome for Android dengan Chrome DevTools. Sedangkan pada perangkat bergerak membutuhkan satu perangkat lunak, yaitu Google Chrome for Android sebagai peramban pada perangkat bergerak.

b. Perangkat Keras

Pada Penelitian ini membutuhkan total tiga

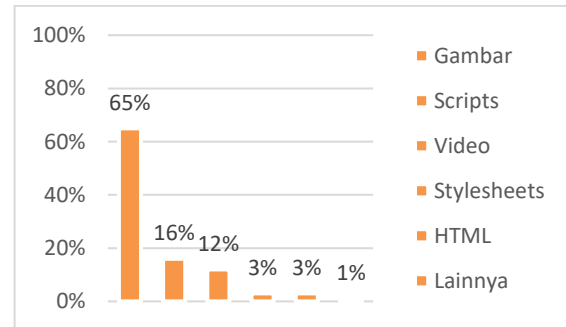
perangkat keras, yaitu komputer yang digunakan untuk melihat data-data hasil pengujian, telepon genggam pintar sebagai perangkat bergerak dan kabel USB-C – USB 2.0/3.0 sebagai penghubung antara komputer dengan perangkat bergerak.

c. Lingkungan Pengujian

Pada penelitian ini membutuhkan beberapa konfigurasi pada lingkungan pengujian. Konfigurasi tersebut berupa area jaringan yaitu jaringan lokal, status koneksi jaringan yaitu terhubung (*online*), kecepatan akses yaitu 5.9 Mbps dan opsi pengelolaan cache yaitu diizinkan (*enabled*). Area jaringan dibutuhkan untuk mengetahui lingkup jaringan yang digunakan pada saat pengujian. Status koneksi jaringan dibutuhkan untuk menentukan kondisi hubungan peladen dengan perangkat bergerak. Kecepatan akses dibutuhkan untuk menentukan seberapa cepat akses jaringan terhadap proses mengunduh dan mengunggah data dari perangkat ke peladen. Opsi pengelolaan cache dibutuhkan untuk menentukan apakah pengelolaan cache dapat dilakukan oleh peramban. Area jaringan lokal dipilih agar kecepatan akses jaringan dapat stabil selama proses pengujian. Status koneksi jaringan online dipilih agar berkas-berkas dapat diterima oleh perangkat baik pada Progressive Web Apps maupun Mobile Web. Kecepatan akses 5.9 Mbps yang merupakan kecepatan rata-rata akses Internet di Indonesia (Belson, 2016), dikarenakan apabila kecepatan akses adalah kecepatan akses yang disediakan peramban, maka kecepatan akses akan terlalu cepat sehingga akan menyulitkan dalam menganalisis dan melihat perbandingan dari dua tipe aplikasi yang diuji. Opsi pengelolaan cache diizinkan agar Mobile Web dapat mengelola cache seperti pada Progressive Web Apps, akan tetapi pengelolaannya diserahkan pada peramban dan sistem operasi.

3.2. Rancangan Objek Kajian

Aplikasi-aplikasi yang diuji pada penelitian ini dirancang sesuai dengan proporsi dari rata-rata berkas penyusun aplikasi *web* saat ini. Adapun proporsi tersebut dapat dilihat dalam Gambar 1.



Gambar 1. Proporsi berkas web

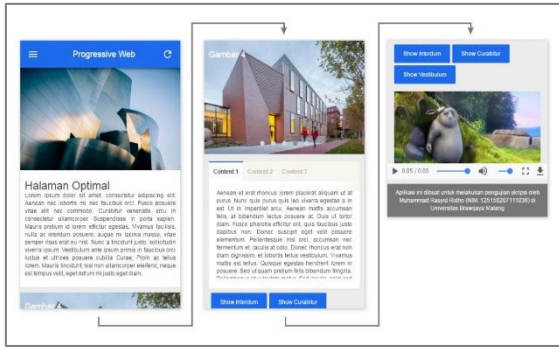
Perancangan struktur berkas aplikasi dilakukan pada masing-masing halaman yang ada. Terdapat tiga halaman pada masing-masing aplikasi, yaitu halaman optimal, sedang dan berat. Penentuan ukuran halaman optimal berdasarkan survei yang dilakukan mengenai seberapa besar ukuran halaman yang baik, yaitu sebesar 1000 kB, lalu untuk halaman berat berdasarkan proyeksi ukuran halaman untuk tahun 2017 (Everts, 2017). Sedangkan halaman sedang merupakan nilai tengah yang juga merupakan rata-rata ukuran halaman *web* pada tahun 2016 (Cremin, 2016).

Antarmuka halaman aplikasi dibagi menjadi dua bagian, yaitu kerangka halaman dan konten halaman. Bentuk dari kerangka halaman mengikuti pedoman Application Shells pada PWA. Bagian kerangka halaman pada aplikasi-aplikasi yang diuji dibagi menjadi tiga blok, yaitu blok header, blok sidebar serta blok konten. Konten halaman tersusun dari beberapa blok. Terdapat tiga jenis blok pada bagian konten halaman, yaitu blok tulisan, blok gambar dan blok video dengan kegunaannya masing-masing yang dapat dilihat pada Tabel 1.

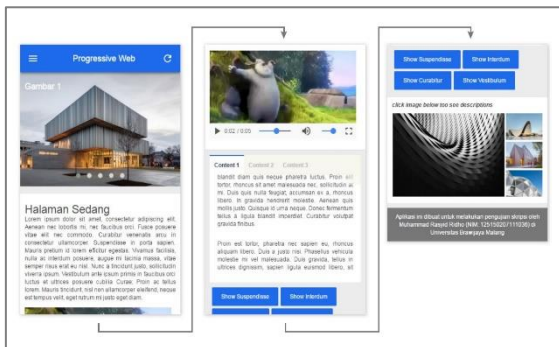
Tabel 1. Blok Antarmuka

No	Blok	Kegunaan
1	Teks	Menampilkan teks pada halaman
2	Gambar	Menampilkan satu gambar pada halaman
3	Video	Menampilkan satu video pada halaman

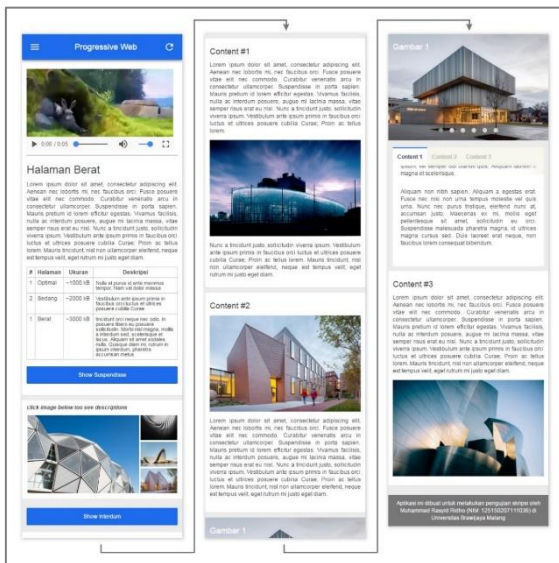
Adapun antarmuka halaman optimal dapat dilihat dalam Gambar 2, sedangkan antarmuka halaman sedang dapat dilihat dalam Gambar 3 dan antarmuka halaman berat dapat dilihat dalam Gambar 4.



Gambar 2. Antarmuka Halaman Optimal



Gambar 3. Antarmuka Halaman Sedang



Gambar 4. Antarmuka Halaman Berat

3.3. Pengujian

a. Variabel Pengujian

Terdapat empat variabel pengujian pada penelitian ini, yaitu tipe aplikasi, fungsionalitas aplikasi, pengaksesan halaman aplikasi dan ukuran *cache*. Variabel-variabel ini akan digunakan sebagai acuan dalam membuat skenario pengujian

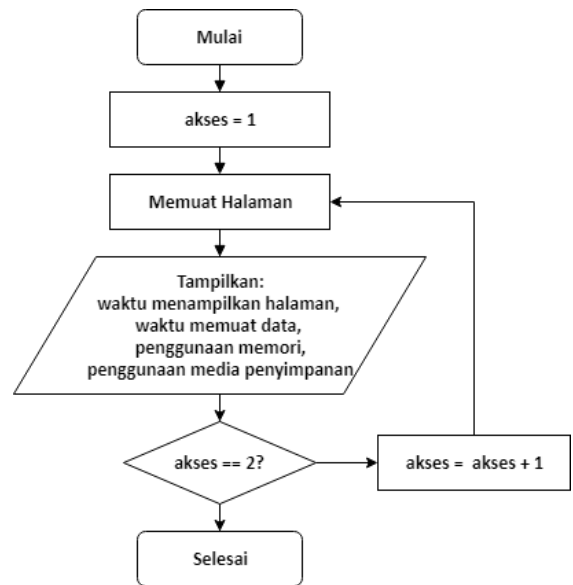
1. Tipe Aplikasi: Progressive Web Apps, Mobile Web.
2. Fungsionalitas Aplikasi: Halaman optimal, halaman sedang, halaman berat.

3. Pengaksesan Halaman Aplikasi: Pengaksesan ke-1, pengaksesan ke-2.
4. Ukuran Berkas Cache: Berkas *cache* dengan persentase menggunakan angka dengan inkremental 10 yang dimulai dari 0 hingga 100 dari total berkas halaman.

b. Skenario Pengujian

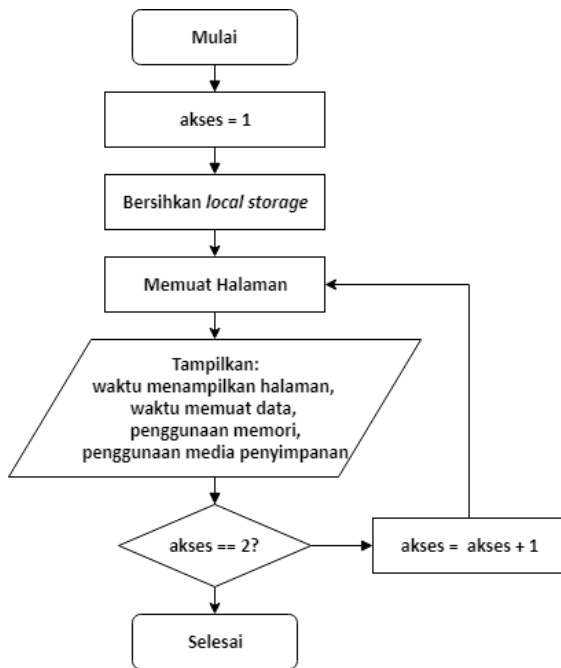
Dalam penelitian ini terdapat 2 skenario, yaitu:

1. Skenario 1 dengan variabel pengujian berupa tipe aplikasi, pengaksesan halaman aplikasi dan fungsional aplikasi (halaman optimal, halaman sedang, halaman berat). Adapun diagram alir dari tahapan-tahapan yang ada pada skenario 1 dijabarkan dalam Gambar 5.



Gambar 5. Diagram Alir Skenario 1

2. Skenario 2 dengan variabel pengujian berupa tipe aplikasi, pengaksesan halaman aplikasi dan fungsional aplikasi (halaman berat). Adapun diagram alir dari tahapan-tahapan yang ada pada skenario 2 dijabarkan dalam Gambar 6.



Gambar 6. Diagram Alir Skenario 2

4. ANALISIS

4.1. Hasil Pengujian

Setelah pengujian dilakukan, data-data hasil pengujian dijabarkan berdasarkan skenario pengujian yang dilakukan. Terdapat dua satuan nilai pada hasil pengujian penelitian ini, yaitu satuan waktu yang dijabarkan dalam milidetik (mdtk) dan satuan ukuran berkas yang dijabarkan dalam *kiloBytes* (kB). Terdapat beberapa nama variabel pengujian dan singkatannya yang digunakan dalam tabel hasil pengujian, yaitu Hal untuk Halaman, (#) untuk Urutan Pengaksesan, WMH untuk Waktu Menampilkan Halaman, WMD untuk Waktu Memuat Data, SW untuk penggunaan memori pada Service Worker, TM untuk Total Memori, MP untuk Media Penyimpanan.

a. Hasil Pengujian Skenario 1

Hasil pengujian skenario 1 dibagi menjadi 2 tabel, yaitu pengujian pada Mobile Web yang dijabarkan pada Tabel 2 dan pengujian pada PWA yang dijabarkan pada Tabel 3.

Tabel 2. Hasil Pengujian Skenario 1 – Mobile Web

Hal	#	WMH (mdtk)	WMD (mdtk)	TM (kB)	MP (kB)
Optimal	1	1761	1254	5146	0
	2	1507	1200	5143	0
Sedang	1	2898	2673	5230	0

Berat	2	2662	2456	5235	0
	1	4305	3864	6167	0
	2	3779	3707	6170	0

Tabel 3. Hasil Pengujian Skenario 1 - PWA

Hal	#	WMH (mdtk)	WMD (mdtk)	MH (kB)	SW (kB)	TM (kB)	MP (kB)
Optimal	1	1898	1613	5170	2350	7520	67
	2	1879	1844	5196	2391	7587	68
Sedang	1	2971	2763	5222	2350	7572	94
	2	2568	2938	5276	2392	7668	95
Berat	1	4374	3856	6219	2350	8569	124
	2	3506	4393	6233	2392	8625	126

b. Hasil Pengujian Skenario 2

Meski pada skenario 2 hanya difokuskan pada variabel ukuran berkas *cache* pada halaman optimal PWA, demi membantu dalam membandingkan dua tipe aplikasi dan mempermudah memahami analisis penelitian nanti, ditambahkan juga hasil pengujian halaman optimal dari Mobile Web secara umum. Oleh karena itu hasil pengujian skenario 2 dibagi menjadi dua tabel, yaitu Tabel 4 yang menjabarkan hasil pengujian halaman optimal dari Mobile Web secara umum dan Tabel 5 yang menjabarkan hasil pengujian halaman optimal dari PWA secara lebih mendetail.

Tabel 4. Hasil Pengujian Skenario 2 – Mobile Web

#	WMH (mdtk)	WMD (mdtk)	TM (kB)	MP (kB)
1	1761	1254	5146	0
2	1507	1200	5143	0

Tabel 5. Hasil Pengujian Skenario 2 – PWA

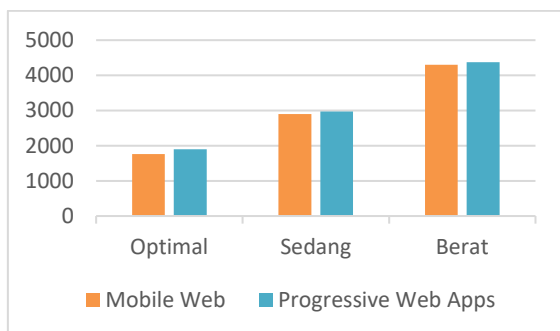
Cache #	#	WMH (mdtk)	WMD (mdtk)	MH (kB)	SW (kB)	TM (kB)	MP (kB)
0%	1	1915	1285	5212	2304	7516	11
	2	1966	1520	5183	2356	7539	11
10%	1	1912	1284	5199	2350	7549	115
	2	1889	1578	5195	2393	7588	145
20%	1	1934	1268	5195	2351	7546	216
	2	1852	1588	5192	2394	7586	246
30%	1	1926	1282	5193	2351	7544	317
	2	1790	1561	5193	2395	7588	347
40%	1	1961	1287	5194	2351	7545	418
	2	1748	1532	5197	2396	7593	448

50%	1	1962	1287	5197	2350	7547	519
	2	1687	1553	5200	2397	7597	549
60%	1	1963	1293	5198	2350	7548	621
	2	1629	1552	5200	2398	7598	651
70%	1	1987	1226	5202	2350	7552	722
	2	1330	1581	5200	2399	7599	752
80%	1	1999	1286	5202	2350	7552	823
	2	1296	1527	5199	2404	7603	853
90%	1	1970	1239	5198	2350	7548	924
	2	1246	1946	5197	2399	7596	954
100%	1	1990	1267	5202	2351	7553	1026
	2	1177	1939	5204	2408	7612	1056

4.2. Analisis Hasil Pengujian

a. Analisis Waktu Respon

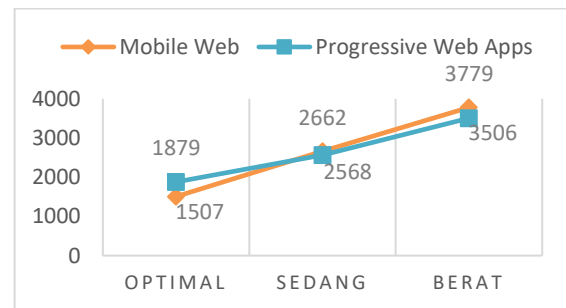
Pada hasil pengujian skenario 1, dua kolom yang merepresentasikan hasil performa dari waktu respon yaitu Waktu Tampil Halaman dan Waktu Muat Data. Untuk kolom Waktu Tampil Halaman, hasil pengujian menunjukkan bahwa pada pengaksesan pertama dengan ukuran berkas halaman optimal, sedang maupun berat, Mobile Web masih lebih cepat dari PWA. Hal ini dikarenakan ada beberapa proses-proses tambahan yang dilakukan oleh PWA dan tidak pada Mobile Web. Contohnya pendaftaran Service Worker, penyimpanan Application Shell pada *cache* dan lain sebagainya. Walaupun demikian, tidak ada selisih perbedaan yang signifikan antara Mobile Web dan PWA. Adapun grafik yang menggambarkan hal ini dapat dilihat dalam Gambar 7.



Gambar 7. WMH Akses ke-1 Skenario 1

Pada pengaksesan ke-2, Mobile Web kembali unggul pada halaman optimal. Namun, tidak pada halaman sedang dan halaman berat, dimana PWA menunjukkan waktu yang lebih cepat. Pada halaman optimal Mobile Web sekitar 300 mdtk lebih cepat dari PWA. Sedangkan pada halaman sedang, PWA sekitar 100 mdtk lebih cepat dari Mobile Web; pada

halaman berat, PWA kembali unggul dengan waktu sekitar 200 mdtk lebih cepat dari Mobile Web. Melihat selisih kecepatan antara dua tipe aplikasi ini, terlihat sebuah tren yang menunjukkan semakin besar ukuran berkas halaman *web* maka semakin baik pula performa dari PWA dibandingkan dengan Mobile Web, dimana pada halaman sedang dan berat PWA justru lebih unggul dari Mobile Web. Namun, belum diketahui sampai ukuran berkas berapa tren ini akan terus berlanjut. Adapun grafik yang menunjukkan tren ini dapat dilihat dalam Gambar 8.



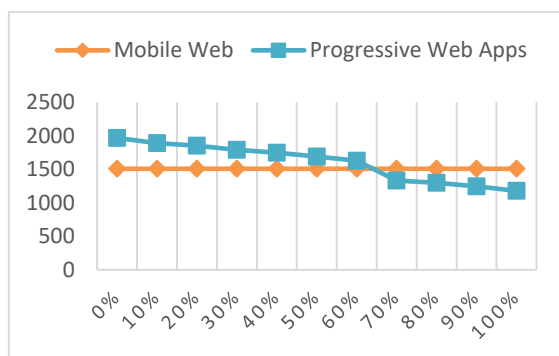
Gambar 8. Tren WMH Akses ke-2 Skenario 1

Berbeda dengan Waktu Tampil Halaman yang menunjukkan tren positif terhadap PWA, kolom Waktu Muat Data pada pengujian skenario 1 menunjukkan bahwa PWA lebih lambat dalam memuat data-data yang dibutuhkan, dalam hal ini adalah berkas-berkas halaman. Mobile Web unggul pada seluruh aspek baik pada pengaksesan pertama dan lebih baik lagi pada pengaksesan kedua. Berbeda dengan PWA yang justru membutuhkan waktu yang lebih lama pada pengaksesan kedua dibandingkan pada pengaksesan pertama. Salah satu hal yang memungkinkan sebagai penyebab akan hal ini adalah adanya pemanggilan berkas-berkas *cache* yang tersedia di media penyimpanan pada saat halaman dimuat pada pengaksesan kedua. Oleh karena itu, terdapat kemungkinan hal ini diakibatkan oleh kecepatan membaca atau menulis dari media penyimpanan pada perangkat uji atau memang hanya kekurangan yang dimiliki PWA.

Pada hasil pengujian skenario 2, hasil performa dari waktu respon juga direpresentasikan oleh dua kolom, yaitu Waktu Tampil Halaman dan Waktu Muat Data. Waktu Tampil Halaman pada PWA secara garis besar lebih lambat sekitar 200 mdtk diseluruh ukuran *cache* yang diujikan pada pengaksesan pertama. Sama halnya dengan pengujian skenario 1, ini disebabkan karena PWA membutuhkan proses-proses tambahan yang dilakukan diawal

pengaksesan halaman. Namun, yang perlu diperhatikan adalah kisaran hasil yang sama Waktu Tampil Halaman pada pengaksesan pertama, yaitu dikisaran 1900 mdtk atau 1,9 detik. Hal ini menunjukkan bahwa Waktu Tampil Halaman tidak terpengaruh oleh besaran data yang ingin dijadikan *cache*. Karena mungkin penyimpanan *cache* kedalam media penyimpanan dilakukan setelah pemuatan berkas selesai yang direpresentasikan oleh Waktu Muat Data.

Pada pengaksesan kedua oleh PWA, Waktu Tampil Halaman kembali menunjukkan sebuah tren seperti pada skenario 1. Namun tren ini bukan terhadap ukuran berkas halaman tetapi ukuran berkas *cache*, dalam kata lain semakin besar ukuran berkas yang dijadikan *cache* pada suatu halaman semakin baik pula performa pada Waktu Tampil Halaman. Namun, perlu dicatat bahwa apabila dibandingkan dengan Mobile Web, performa PWA pada Waktu Tampil Halaman baru bisa melampaui performa Mobile Web pada persentase *cache* 60% keatas. Selain itu, selisih terbesar yaitu pada persentase *cache* 100% berkisar 300 mdtk dengan Mobile Web. Adapun grafik yang menunjukkan tren ini dapat dilihat dalam Gambar 9.



Gambar 9. Tren WMH Akses ke-2 Skenario 2

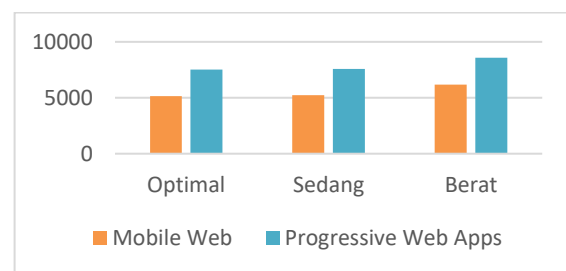
Masih pada hasil pengujian skenario 2, Waktu Muat Data berada diangka yang seragam pada hampir semua persentase *cache* yang diujikan, yakni dikisaran 1,2 detik untuk pengaksesan pertama dan 1,5 detik pada pengaksesan kedua. Namun terdapat anomali pada persentase *cache* 90% dan 100% dimana pada pengaksesan kedua berada pada kisaran 1.9 detik. Hal ini mungkin saja disebabkan dimuatnya tipe berkas video dari *cache*, yang mana pada persentase 0% - 80% berkas video tidak dijadikan berkas *cache*. Dibandingkan dengan Mobile Web, Waktu Muat Data untuk pengaksesan pertama pada PWA masih sama pada kisaran 1,2 detik. Sedangkan pada

pengaksesan kedua, PWA lebih lambat 300 mdtk yakni dikisaran 1,5 detik.

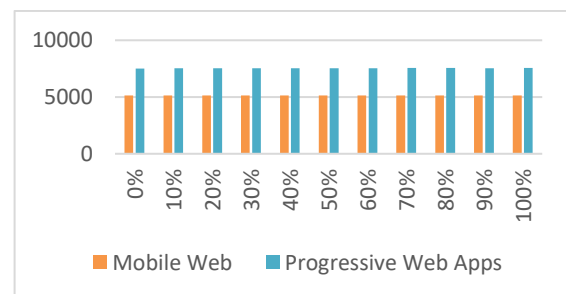
Dari analisis ini dapat disimpulkan bahwa PWA akan lebih baik diimplementasikan pada aplikasi web yang memiliki ukuran berkas yang cukup banyak dan ekspektasi pengaksesan yang lebih dari satu kali, seperti aplikasi *web e-commerce* yang memiliki banyak fungsionalitas dan gambar, portal berita yang biasanya sering dikunjungi, dsb.

b. Analisis Penggunaan Memori

Pada pengujian skenario 1 dan pengujian skenario 2, total penggunaan memori oleh Mobile Web dan PWA menunjukkan pola yang sama. Dimana penggunaan memori pada PWA selalu lebih besar dibandingkan dengan penggunaan memori pada Mobile Web. Hal ini dikarenakan pada PWA terdapat dua proses yang berjalan di memori perangkat, yaitu proses pada halaman *web* dan proses *Service Worker*. Selain itu, selisih penggunaan memori baik pada tiga fungsional halaman yang diujikan pada skenario 1 maupun seluruh persentase *cache* yang diujikan pada skenario 2 juga sama, yakni dikisaran 2300 kB. Adapun grafik yang menunjukkan perbandingan penggunaan memori pada skenario 1 dapat dilihat dalam Gambar 10, sedangkan perbandingan penggunaan memori pada skenario 2 dapat dilihat dalam Gambar 11.



Gambar 10. Penggunaan Memori Skenario 1



Gambar 11. Penggunaan Memori Skenario 2

Pada pengujian skenario 1, selisih penggunaan memori pada halaman optimal dan sedang tidak terlampau jauh yakni dikisaran 100

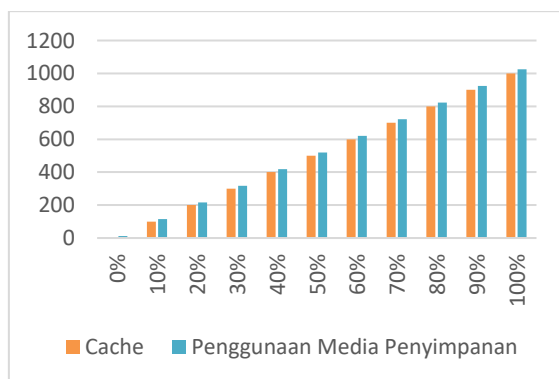
kB baik pada Mobile Web maupun PWA. Tetapi, hal ini tidak terjadi pada selisih antara halaman sedang dan berat dimana selisih berada dikisaran 1000 kB. Tidak diketahui secara pasti alasan dibalik hal ini. Terdapat kemungkinan hal ini disebabkan lagi oleh tipe-tipe berkas yang dimuat, khususnya pada halaman berat. Dikarenakan hal ini terjadi baik pada Mobile Web maupun PWA.

Tidak ada selisih perbedaan penggunaan memori yang signifikan pada pengaksesan pertama dan kedua, dimana rata-rata selisih perbedaan hanya dikisaran kurang dari 50 kB baik pada Mobile Web maupun PWA. Hal ini menunjukkan bahwa penggunaan memori tidak terpengaruh oleh urutan pengaksesan, khususnya pada PWA yang memuat berkas *cache* terlebih dahulu pada pengaksesan kedua dan seterusnya.

Dari hasil ini dapat dikatakan bahwa pemilihan tipe aplikasi PWA atau Mobile Web untuk diimplementasikan akan kurang efektif apabila hanya melihat dari performa penggunaan memori, dikarenakan selisih penggunaan pada dua tipe aplikasi yang diuji tidak terlalu signifikan

c. Analisis Penggunaan Media Penyimpanan

Pada pengujian skenario 1 dan pengujian skenario 2, total penggunaan media penyimpanan oleh Mobile Web tidak berubah, yakni 0 kB dalam arti tidak menggunakan ruang media penyimpanan. Sedangkan pada PWA total penggunaan media penyimpanan menyesuaikan dengan besaran ukuran berkas yang dijadikan *cache* ditambah dengan berkas-berkas lain. Hal ini digambarkan dan dapat dilihat dalam Gambar 12. Berkas-berkas lain tersebut adalah *service-worker.js* dan *app.js* yang menjadi pondasi dari tipe aplikasi PWA.



Gambar 12. Penggunaan Media Penyimpanan Skenario 2

Pada hasil pengujian skenario 1, terdapat selisih total penggunaan media penyimpanan pada pengaksesan ke 1 dan ke 2 yang hanya berkisar 1 kB. Tidak diketahui apa yang menyebabkan perbedaan ini, tetapi perbedaan ini tidaklah terlalu signifikan apabila melihat total penggunaan media penyimpanan yang digunakan. Sebagai catatan, total penggunaan media penyimpanan pada PWA berkisar diangka 67 kB pada halaman optimal, 94 kB pada halaman sedang dan 124 kB pada halaman berat.

Berbeda dengan hasil pengujian skenario 1, pada hasil pengujian skenario 2 terdapat selisih yang lebih besar pada total penggunaan media penyimpanan pada pengaksesan ke 1 dan ke 2. Selisih tersebut sebesar 30 kB yang mana terjadi pada hampir seluruh persentase *cache* yang diujikan, kecuali pada persentase *cache* 0%. Pada persentase *cache* 0% tidak terdapat selisih antara pengaksesan ke 1 dan ke 2, yakni hanya 11 kB yang merupakan berkas *service-worker.js* dan *app.js*. Perbedaan pada persentase *cache* 0% dengan persentase lainnya dikarenakan tidak adanya berkas yang dijadikan *cache* pada persentase tersebut.

Selisih antara pengaksesan pertama dan kedua baik pada skenario 1 maupun 2 terlihat ganjil, karena secara logika pada pengaksesan pertama seluruh berkas yang dijadikan *cache* sudah disimpan pada media penyimpanan dan seharusnya pada pengaksesan kedua berkas hanya tinggal dipanggil kembali dari media penyimpanan, tidak perlu menyimpan *cache* yang lain; dalam arti total *cache* pada media penyimpanan seharusnya tetap sama. Tetapi yang perlu dicatat bahwa penggunaan media penyimpanan pada PWA hampir berbanding lurus dengan berkas-berkas yang dijadikan *cache* pada suatu halaman.

Oleh karena itu, pengetahuan yang lebih akan batasan-batasan penggunaan media penyimpanan permban dan kebijaksanaan akan pemilihan berkas yang akan dijadikan *cache* akan sangat membantu dalam menghasilkan baik atau buruknya performa pada PWA.

5. KESIMPULAN

Berdasarkan penelitian yang dilakukan, dapat disimpulkan beberapa hal sebagai berikut:

1. Performa terkait waktu respon menyesuaikan dengan ukuran berkas dan *cache* yang digunakan serta frekuensi pengaksesan 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. Perlu dicatat bahwa performa dari PWA juga akan lebih baik apabila frekuensi pengaksesan yang tidak hanya sesekali saja.

2. Performa terkait penggunaan memori pada Mobile Web lebih kecil dibandingkan dengan PWA. Selisih penggunaan memori antara Mobile Web dan PWA pada penelitian ini tidak terlampau jauh, yakni sekitar 2300 kB. Dimana selisih tersebut disebabkan oleh adanya proses tambahan pada PWA, yaitu Service Worker.
3. Performa terkait penggunaan media penyimpanan pada Mobile Web lebih unggul dibandingkan dengan Progressive Web App, dimana pada Mobile Web Tidak Menggunakan Ruang Penyimpanan Sama Sekali. Namun, Keunggulan Dari Mobile Web tidak terlampau jauh dari PWA menyesuaikan dengan ukuran berkas *cache* yang disimpan. Pada PWA penggunaan media penyimpanan terdiri dari berkas *cache* yang disimpan ditambah beberapa berkas fungsional PWA yang ukurannya terbilang kecil.

channel/offline-storage-for-progressive-web-apps-70d52695513c> [Diakses 14 Maret 2017]

- Russel, A., 2015. Progressive Web Apps: *Escaping Tabs Without Losing Our Soul*. [online] Tersedia di: <<https://infrequently.org/2015/06/progressive-apps-escaping-tabs-without-losing-our-soul/>> [Diakses 1 Februari 2017]

DAFTAR PUSTAKA

- AppDynamics, 2014. *Mobile App Performance Explained*. [online] Tersedia di: <<https://kapost-files-prod.s3.amazonaws.com/published/54dd1e53715cba88750000e4/white-paper-mobile-app-performance-explained.pdf>> [Diakses 3 Maret 2017]
- Belson, D., 2016. *Akamai's [state of the internet] – Q2 2016 Report, 9(2), p.43*.
- Cremin, R., 2016. The web is Doom. Tersedia di: <<https://mobiforge.com/research-analysis/the-web-is-doom>> [Diakses 6 April 2017]
- Everts, T., 2015. Page bloat update: The average web page is more than 2 MB in size. [online] Tersedia di: <<https://www.soasta.com/blog/page-bloat-average-web-page-2-mb/>> [Diakses 22 Maret 2017]
- Osmani, A., 2016. *Offline Storage for Progressive Web Apps*. [online] Tersedia di: <<https://medium.com/dev->