

IMPLEMENTASI PROGRESSIVE WEB APPS UNTUK SISTEM PENGELOLAAN POTENSI DESA WISATA KALI OPAK TUJUH BULAN

Erna Hudianti¹⁾, Didik Maulana²⁾, Muhammad Agung Nugroho³⁾

^{1,2,3)} Informatika, Universitas Teknologi Digital Indonesia
email : ernahudi@utdi.ac.id¹⁾, didikmaulana48@gmail.com²⁾, magung.n@utdi.ac.id^{3*)}

Abstraksi

Simpel Dawis (Sistem Informasi Pengelolaan Potensi Desa Wisata) merupakan sebuah sistem informasi yang digunakan di Wisata Kali Opak Tujuh Bulan untuk media informasi kepada masyarakat dan pengunjung tentang wisata kali opak tujuh bulan. Kondisi jaringan yang berbeda-beda terutama wisata alam membutuhkan sebuah teknologi yang dapat memudahkan pengunjung dalam mengakses website kapan pun dan dimana pun, bahkan dalam kondisi offline. Progressive Web Apps (PWA) adalah konsep pengembangan aplikasi berbasis web yang mencakup teknologi terbaru dari browser seperti web apps manifest, service worker dan app shell. Konsep PWA diharapkan mampu memberikan pengalaman terbaik dalam menggunakan suatu aplikasi web walaupun dalam kondisi jaringan yang minim atau offline sekalipun. Service worker sebagai pengatur request dan respon pada sebuah aplikasi web dapat diatur sesuai kebutuhan. Pada studi ini, service worker menggunakan strategi cache first then network, cache fallback untuk menghindari error pada kerangka kerja Laravel dan tetap mendapat respon yang cepat ketika web dibuka. Penerapan konsep PWA khususnya service worker meningkatkan performa website Simpel Dawis pada saat memuat halaman dan berjalan secara offline. Selain itu memberikan referensi tentang perbandingan web native dan web progresif.

Kata Kunci :

Laravel, Progressive Web Apps, Service Worker, Web App Manifest

Abstract

Simpel Dawis (Tourism Village Potential Management Information System) is an information system used at the Desa Wisata Kali Opak Tujuh Bulan for information media to the people of Wisata Kali Opak. Various network conditions, specifically outdoor tourism, require a technology that can make it easier for tourists to access websites anytime and anywhere, even offline. Progressive Web Apps (PWA) is a web-based application development concept that contains the latest technologies from browsers such as web apps manifest, service workers and app shells. The PWA concept is desired to provide the best experience using a web application, even in the minimal network connection or offline network conditions. Service workers as request and response regulators in a web application can be arranged as needed. In this study, service workers use a cache first, then a network cache fallback strategy to avoid errors in the Laravel framework and still get a fast response when the web is opened. Applying the PWA concept, especially for service workers, improves the performance of the Simple Dawis website when loading pages and running offline. In addition, it provides references on the comparison of the native and progressive web.

Keywords:

Laravel, Progressive Web Apps, Service Worker, Web App Manifest

Pendahuluan

Obyek wisata yang memiliki potensi besar, penting untuk dikelola dengan baik. Salah satunya perlu sebuah media informasi untuk mempublikasikan obyek wisata tersebut. Desa Wisata merupakan bentuk integrasi antara atraksi, akomodasi dan fasilitas pendukung yang disajikan dalam suatu struktur kehidupan masyarakat yang memiliki kesatuan dengan tata cara dan budaya yang berlaku [1]. Dalam menggunakan website untuk kepentingan wisata dalam pengembangan pemasaran di Desa wisata sangat memungkinkan untuk dilakukan, jika memperhitungkan mengenai jangkauan dan sebaran media online lebih besar dibandingkan media pemasaran dengan model konvensional [2]. Sistem

informasi pariwisata dapat digunakan oleh pengunjung web untuk melihat informasi kesenian yang ada di suatu objek pariwisata. Pengembangan sistem informasi dapat dijadikan alat bantu dalam pengelolaan data objek dan layanan wisata yang ada di suatu daerah [3]. Sistem yang dikembangkan dapat diakses menggunakan *device* seperti *desktop* dan *mobile* selama terhubung dengan koneksi internet. Minimnya pemahaman terhadap teknologi menjadi hal yang menghambat pengelolaan data pada objek wisata khususnya desa wisata. Sebelum merencanakan sistem informasi objek wisata, desa wisata kali opak tujuh bulan masih mengelola data secara manual dalam bentuk *spreadsheet* dan belum dapat berjalan secara

terintegrasi. Pengembangan aplikasi ini bertujuan untuk melakukan uji coba dalam membuat sistem aplikasi dengan teknologi *Progressive Web Apps* (PWA). Penulis membangun sebuah sistem bernama Simpel Dawis (Sistem Informasi Pengelolaan Potensi Desa Wisata) di Wisata Kali Opak Tujuh Bulan yang terletak di Dusun Dalem Tamarmartani. Simpel Dawis ini terdiri dari satu aplikasi berbasis mobile yang digunakan oleh pengelola untuk membantu pengelolaan wisata kali opak tujuh bulan dan satu aplikasi berbasis website yang digunakan untuk media informasi kepada masyarakat dan pengunjung tentang wisata kali opak tujuh bulan. Website Simpel Dawis tersebut berisi tentang profil obyek wisata, kegiatan-kegiatan yang ada di wisata tersebut baik yang sudah dilaksanakan, yang sedang dilaksanakan maupun yang akan dilaksanakan. Oleh karena itu, website tersebut diwajibkan untuk selalu di-update secara cepat oleh admin agar mampu memberikan informasi yang cepat kepada masyarakat dan pengunjung. Dalam menerima update konten yang ada di dalam website, pengguna harus membuka browser dan masuk ke website tersebut dengan menggunakan perangkat elektronik berupa komputer/laptop atau handphone. Kondisi jaringan yang berbeda-beda terutama wisata alam membutuhkan sebuah teknologi yang dapat memudahkan pengunjung dalam mengakses website kapan pun dan dimana pun, bahkan dalam kondisi offline sekali pun. Selain itu, performa website yang baik juga berpengaruh terhadap website tersebut agar lebih optimal.

Progressive Web Apps (PWA) diharapkan mampu mengatasi permasalahan yang terjadi di wisata kali opak tujuh bulan tersebut. PWA ini akan memberikan pengalaman baru kepada user dengan mengunjungi sebuah website yang dibangun menggunakan teknologi web modern, namun dapat berlaku seperti mobile apps. Selain itu, website ini juga akan menggunakan teknologi *service worker* yang dapat memberikan fungsionalitas offline, notifikasi, update content, pergantian konektivitas yang dapat berjalan dalam kondisi jaringan yang tidak stabil atau offline sekalipun.

Penelitian ini akan mengimplementasikan *Progressive Web Apps* kepada website wisata kali opak tujuh bulan dan melihat perbedaan antara sebelum dan sesudah menggunakan teknologi ini.

Tinjauan Pustaka

Progressive Web Apps (PWA) [4] merupakan bentuk dari pengembangan menggunakan teknologi web modern, namun dapat berlaku seperti aplikasi yang berjalan pada perangkat mobile apps. Alex Russel dan Frances Berriman pada tahun 2015 menggunakan istilah *Progressive Web Apps* sebagai konsep yang dapat memberikan user experience dalam bentuk *reliabilit*, *speed*, dan user engagement. Pada website atau aplikasi web yang memiliki kemampuan progresif dapat memanfaatkan web browser modern yang memiliki dukungan terhadap

PWA. Aplikasi PWA tidak dikemas (*packaged*) dan *deploy* melalui playstore, melainkan pada suatu layanan aplikasi web yang dibagikan melalui sebuah link yang ditambahkan ke Home Screen bersama dengan izin untuk menerima *notifikasi* dari web tersebut [5].

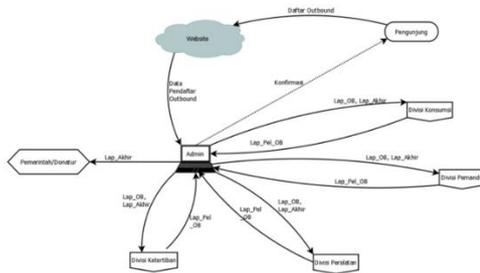
Pada komponen PWA terdapat *service worker* yang salah satu jenis *Web Worker* berupa JavaScript yang dapat bekerja secara background dan tidak mempengaruhi performa dari halaman pada web [6]. *Service Worker* merupakan file JavaScript yang bekerja dari sisi klien dan terpisah dari rangkaian kerja pada browser utama, berfungsi mengalihkan request pada jaringan, melakukan proses cache atau mengambil sumber daya dari cache, dan meneruskan pesan. Dalam teknis *service worker* menyediakan script proxy jaringan di browser web untuk mengelola *HTTP Request* secara terprogram. *Service Worker* terletak antara jaringan dan perangkat mobile untuk meneruskan konten aplikasi. *Service Worker* memanfaatkan mekanisme *cache* secara efisien dan memastikan adanya *error-free* selama periode offline. *Service worker* dapat mengendalikan semua halaman yang ada dalam ruang lingkungannya, walaupun halaman yang mendaftarkan *service worker* untuk pertama kali tidak dapat dikendalikan sampai halaman dapat dimuat kembali. Saat *service worker* berjalan kembali, dapat berada dalam satu dari dua kondisi, baik *service worker* berhenti untuk menghemat memori atau memberikan solusi dengan mengembalikan data dan pesan saat ada yang terjadi dari jaringan atau ada pesan yang dimuat dari sebuah halaman website [7].

Teknologi PWA dan *service worker* dapat digunakan untuk melakukan pelacakan dan pengiriman barang serta laporan pengiriman yang dapat berfungsi secara online dan saat kondisi jaringan dengan dalam kondisi offline. Implementasi ini digunakan pada pelacakan barang PT. Oriflame [8]. Selain itu, teknologi yang sama juga dapat digunakan pada pengembangan aplikasi reservasi homestay di daerah desa wisata tembui, dari hasil pengujian aplikasi dapat berjalan dengan lancar baik saat online ataupun offline [9]. Dalam konsep pengujian dapat dilakukan perbandingan performa antara PWA dan *web native* dengan mengukur dari sisi performance dan page details dengan memanfaatkan tools GTMetrix. Hasil pengujian menunjukkan bahwa PWA memiliki performa yang lebih dari *web native* saat terjadi permasalahan koneksi internet. Selain itu, hasil pengujian PWA dan *service worker* dengan menggunakan lighthouse dan chrome devtools pada platform e-learning menunjukkan peningkatan secara performa saat menampilkan halaman secara offline [10].

Metode Penelitian

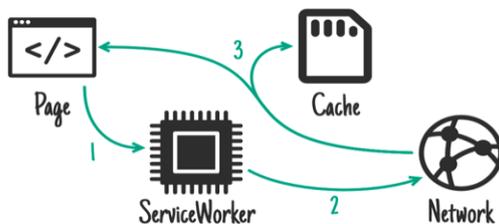
Penelitian ini dilakukan dengan merancang sebuah website untuk desa wisata. Simpel Dawis merupakan

sebuah aplikasi yang digunakan untuk pengelolaan dan media informasi di Desa Wisata Kali Opak Tujuh Bulan. Gambaran hubungan antar aktor Simpel Dawis dapat dilihat pada gambar 1.



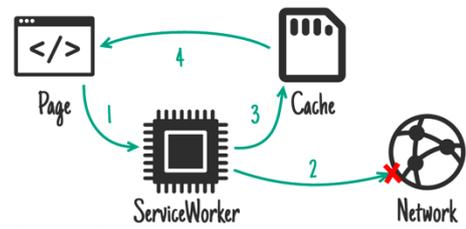
Gambar 1. Gambaran sistem Simpel Darwis

Cara kerja website adalah pengguna melakukan akses pada website berupa URL melalui web browser. Web browser mengirimkan request dalam bentuk HTTP request kepada Web Server melalui lapisan TCP/IP. Selanjutnya, Web Server memberikan respon dengan menyampaikan request yang diminta dengan kondisi jaringan dapat diakses. Ada 3 komponen utama agar semua aplikasi web bisa dikatakan sebagai PWA yaitu : *app shell*, *app manifest* dan *service worker*. Service Worker berperan utama dalam konsep PWA. Service Worker melakukan pekerjaan untuk mengatur *event fetch* dari browser, kemudian service worker memutuskan *request* akan di *forward* menuju server atau menuju *cache* dengan melihat kondisi jaringan online atau offline. Strategi *caching* yang digunakan oleh *service worker* umumnya dengan metode *network first* dan *cache fallback*. Dengan menggunakan strategi ini, *service worker* dapat melakukan pengecekan dalam ketika *network* memberikan respons, dan jika berhasil akan memberikan data terbaru ke halaman browser. Saat terjadi kegagalan, *service worker* mengambil data dari *cache*. Strategi ini dipakai saat membutuhkan data yang selalu baru misalkan respons API, namun membutuhkan tampilan pada halaman web saat jaringan tidak terhubung. *Service Worker* dapat meneruskan setiap request dari halaman web menuju server dalam kondisi online, lalu menduplikasi respons server dan disimpan ke dalam *cache* di browser, kemudian respons server meneruskan kembali ke halaman web seperti yang digambarkan pada Gambar 2 dimana setiap halaman yang diakses oleh pengguna secara otomatis tersedia dalam versi offline-nya, namun pengguna tidak dapat mengirim data ke server.



Gambar 2. Alur service worker dalam kondisi online

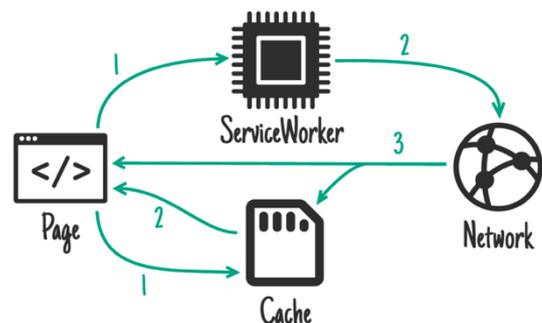
Ketika kondisi *offline* atau tidak ada respons dari server, maka akan mengambil data *cache* dan halaman bisa ditampilkan apabila halaman tersebut sudah ada di dalam *cache* sebelumnya seperti pada Gambar 3.



Gambar 3. Alur service worker dalam kondisi offline

Admin memiliki peranan penting dalam sistem ini, baik untuk aplikasi website ataupun mobile. Website selain media informasi untuk pengunjung mengenai Wisata Kali Opak Tujuh Bulan juga digunakan pengunjung untuk mendaftar sebagai peserta outbound yang merupakan paket wisata yang ditawarkan oleh pengelola Wisata Kali Opak Tujuh Bulan. Aplikasi mobile yang digunakan oleh pengelola untuk memberikan laporan setiap kegiatan yang dilakukan oleh masing-masing divisi salah satunya adalah kegiatan outbound yang nantinya akan menjadi laporan kegiatan kepada pemerintah setempat.

Pengujian service worker perlu dilakukan untuk melihat perbedaan performa dari layanan yang dibuat. Website Simpel Dawis memiliki dua user yaitu admin dan pengunjung. Simple Dawis memiliki beberapa halaman website diantaranya adalah halaman Beranda, Profil, Galeri, Paket Outbound dan Kontak. Website akan diuji dalam beberapa kondisi yaitu sudah atau belum diakses website tersebut sebelumnya, status *service worker*, *cache* halaman dan kondisi jaringan. Model pengujian *service worker* dari *cache* ke *network* diilustrasikan pada gambar 4.



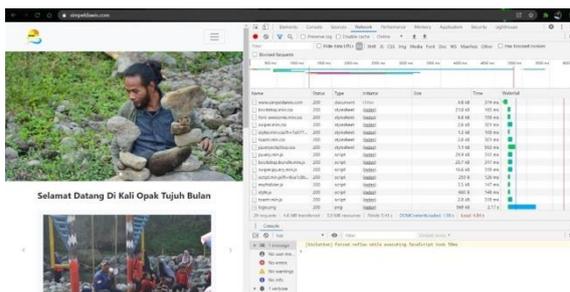
Gambar 4. Alur Service worker dari cache ke network

Proses *cache first then network*, pendekatan ini akan mendapatkan konten di layar secepat mungkin, tetapi tetap menampilkan konten terbaru. Halaman membuat dua permintaan: satu ke *cache* dan satu ke *network*. Halaman akan menampilkan data dari *cache* terlebih dahulu, lalu memperbaharui halaman jika data dari *network* tiba. Proses Pengujian dilakukan dalam beberapa tahapan yaitu: Pengujian

1 dan 2 merupakan website yang belum menggunakan *Service Worker*. Pengujian 3 sampai 6 sudah menggunakan *Service Worker*. Pengujian diatas menggunakan Chrome DevTools dan hasil yang dilihat adalah jumlah *request*, ukuran data yang diterima, waktu memuat halaman dan waktu memuat semua *request*.

Hasil dan Pembahasan

Implementasi sistem merupakan tahap penerapan sekaligus pengujian bagi sistem berdasarkan hasil analisa dan perancangan pada bab sebelumnya, sehingga sistem yang buat benar-benar dapat dioperasikan dan dapat mencapai tujuan yang diharapkan. Sistem Informasi Pengelolaan Potensi Desa Wisata (Simpel Dawis) yang sudah dibuat akan dimodifikasi dengan menambahkan Progressive Web Apps diantaranya teknologi *Web App Manifest*, *Service Worker* dan *App Shell*. Simpel Dawis yang sudah dimodifikasi menggunakan Progressive Web Apps dilakukan pengujian dengan beberapa skenario yang sudah ditentukan pada pembahasan sebelumnya. Gambar 5 merupakan deskripsi skenario pada proses pengujian.



Gambar 5. Tampilan inspect element untuk proses pengujian

Hasil pengujian pertama ketahui ada 29 *request* yang dibuat dari halaman, 4,6 MB data diterima, waktu memuat halaman 4,84 detik dan selesai memuat semua *request* dengan waktu 5,41 detik. Pada hasil pengujian kedua, diketahui ada 8 *request* yang dibuat dari halaman. Seperti halnya website tradisional lain ketika tidak memiliki koneksi atau offline maka halaman akan gagal dimuat. Dilanjutkan pada pengujian ketiga dengan 45 *request* yang dibuat dari halaman, 4,8 MB data diterima, waktu memuat halaman 7,08 detik dan selesai memuat semua *request* dengan waktu 7,51 detik. Sementara itu, pada pengujian keempat dengan 49 *request* yang dibuat dari halaman, 0 B data diterima, waktu memuat halaman 403 Milidetik dan selesai memuat semua *request* dengan waktu 408 Milidetik. Berbeda dengan pengujian kelima yang menggunakan 46 *request* yang dibuat dari halaman, 797 KB data diterima, waktu memuat halaman 633 milidetik dan selesai memuat semua *request* dengan waktu 1,21 detik dan pengujian keenam diketahui ada 51 *request* yang dibuat dari halaman, 0 B data diterima, waktu memuat halaman

427 Milidetik dan selesai memuat semua *request* dengan waktu 437 Milidetik.

Dari ujicoba yang dilakukan memperlihatkan hasil *caching service worker* yang disimpan di *cache storage*. Terdapat dua file variabel seperti yang dibuat pada pembahasan 4.1.2 yaitu file *site-static* dan *site-dynamic*. *Site-static* digunakan untuk *cache* yang bersifat static atau disebut dengan *App Shell* yang berisi HTML, CSS dan JavaScript. *Site-dynamic* berisi file yang bersifat dinamis seperti file dari database atau dokumen lain. Hasil pengujian secara keseluruhan dideskripsikan pada tabel 1 yang menggambarkan kondisi pengujian dan hasil dari pengujian berupa *request*, data yang diterima, waktu untuk memuat halaman dan semua *request*.

Tabel 1. Hasil Pengujian

No	Kondisi	Hasil
1	<ul style="list-style-type: none"> • Kunjungan Halaman: Pertama • <i>Service Worker</i>: tidak ada • <i>Cache</i>: tidak ada • Jaringan: online • Update konten: tidak 	<ul style="list-style-type: none"> • Jumlah <i>request</i>: 29 <i>request</i> • Data diterima: 4,6 MB • Memuat Halaman: 4,84 detik • Memuat semua <i>request</i>: 5,41 detik
2	<ul style="list-style-type: none"> • Kunjungan Halaman: kedua • <i>Service Worker</i>: tidak ada • <i>Cache</i>: tidak ada • Jaringan: offline • Update konten: tidak 	<ul style="list-style-type: none"> • Jumlah <i>request</i>: 0 <i>request</i> • Data diterima: 0 MB • Memuat Halaman: - • Memuat semua <i>request</i>: -
3	<ul style="list-style-type: none"> • Kunjungan Halaman: ketiga • <i>Service Worker</i>: ada • <i>Cache</i>: tidak ada • Jaringan: online • Update konten: tidak 	<ul style="list-style-type: none"> • Jumlah <i>request</i>: 45 <i>request</i> • Data diterima: 4,8 MB • Memuat Halaman: 7,08 detik • Memuat semua <i>request</i>: 7,51 detik
4	<ul style="list-style-type: none"> • Kunjungan Halaman: keempat • <i>Service Worker</i>: ada • <i>Cache</i>: ada • Jaringan: online • Update konten: ada 	<ul style="list-style-type: none"> • Jumlah <i>request</i>: 49 <i>request</i> • Data diterima: 0 MB • Memuat Halaman: 403 milidetik • Memuat semua <i>request</i>: 408 detik
5	<ul style="list-style-type: none"> • Kunjungan Halaman: kelima • <i>Service Worker</i>: ada • <i>Cache</i>: ada • Jaringan: online 	<ul style="list-style-type: none"> • Jumlah <i>request</i>: 46 <i>request</i> • Data diterima:

	<ul style="list-style-type: none"> Update konten: ada 	797 MB <ul style="list-style-type: none"> Memuat Halaman: 633 milidetik Memuat semua request: 1,21 detik
6	<ul style="list-style-type: none"> Kunjungan Halaman: keenam Service Worker: ada Cache: ada Jaringan: offline Update konten: tidak 	<ul style="list-style-type: none"> Jumlah request: 51 request Data diterima: 0 MB Memuat Halaman: 427 milidetik Memuat semua request: 437 milidetik

Hasil pengujian diatas terdapat enam skenario pengujian. Pengujian pertama dan kedua merupakan kondisi website sebelum menggunakan PWA dalam hal ini *service worker*. Simpel Dawis seperti web native pada umumnya, ketika diakses dengan kondisi jaringan online atau stabil maka akan berjalan dengan normal. Pada saat diakses dengan kondisi jaringan offline maka akan menampilkan pesan *error* atau website tidak bisa diakses.

Kesimpulan dan Saran

Berdasarkan hasil analisa, perancangan, implementasi dan pengujian yang telah dilakukan pada penelitian ini didapatkan beberapa kesimpulan yaitu Simpel Dawis dengan menerapkan teknologi *Progressive Web Apps* mampu di-install di perangkat mobile dan desktop dan mampu menampilkan halaman pada saat kondisi jaringan tidak stabil atau offline. Pada hasil pengujian menunjukkan bahwa website Simpel Dawis yang menggunakan teknologi *Progressive Web Apps* pada saat pertama kali diakses oleh user memiliki waktu 31.6% lebih lambat dari *web native*, tetapi 88,5% lebih cepat setelah menandakan *service worker* dan memiliki *cache*.

Simpel Dawis yang telah menerapkan teknologi *Progressive Web Apps* dan melalui pengujian yang telah dilakukan, terdapat beberapa saran yang dapat diberikan untuk pengembangan dan perbaikan di masa yang akan datang yaitu menambahkan fitur *Push Notification* agar setiap terjadi perubahan data dapat mengirimkan pesan kepada pengguna dan menerapkan PWA pada halaman admin.

Daftar Pustaka

[1] R. R. Rosa, R. Salman, S. Winarsi, and W. Prihatiningtyas, "Pengembangan Potensi Desa Wisata Di Masa Pandemi (Studi Di Desa Sukobendu, Lamongan)," *J. Dedik. Huk.*, vol. 1, no. 3, pp. 313–327, 2021.

[2] A. Damayanti, A. Nugroho, and S. Windarti, "PENINGKATAN KAPASITAS KELOMPOK

DESA WISATA WUKIRSARI DALAM PEMASARAN ONLINE MENGGUNAKAN SOSIAL MEDIA MARKETING," *J. Pengabd. Masy. - Teknol. Digit. Indones.*, vol. 1, no. 1, p. 15, Mar. 2022, doi: 10.26798/jpm.v1i1.566.

- [3] Y. Enga and E. Faizal, "SISTEM INFORMASI PARIWISATA KABUPATEN NAGEKEO BERBASIS WEBSITE," *SPEED - Sentra Penelit. Eng. Dan Edukasi*, vol. 14, no. 3, pp. 68–74, 2022.
- [4] D. Sheppard, "Introduction to Progressive Web Apps," in *Beginning Progressive Web App Development: Creating a Native App Experience on the Web*, D. Sheppard, Ed. Berkeley, CA: Apress, 2017, pp. 3–10. doi: 10.1007/978-1-4842-3090-9_1.
- [5] H. Santoso, *Membangun Aplikasi Mobile dengan PWA (Progressive Web App)*. Lokomedia, 2019.
- [6] C. Love, *Progressive Web Application Development by Example: Develop fast, reliable, and engaging user experiences for the web*. Packt Publishing Ltd, 2018.
- [7] M. Gaunt, "Service workers: an introduction," *Hacker Bits*, no. 12, pp. 6–15, 2018.
- [8] D.-155410082 Wulandari, "PROGRESSIVE WEB APPS (PWA) UNTUK APLIKASI PELACAKAN PENGIRIMAN BARANG (STUDI KASUS JASA PENGIRIMAN PT.ORIFLAME)," skripsi, STMIK AKAKOM YOGYAKARTA, 2019. Accessed: Jan. 01, 2023. [Online]. Available: <https://eprints.utdi.ac.id/8454/>
- [9] R. S. Purba, "IMPLEMENTASI PROGRESSIVE WEB APPS DENGAN TEKNOLOGI SERVICE WORKER (STUDI KASUS KELUARGA MAHASISWA STMIK AKAKOM YOGYAKARTA)," skripsi, STMIK AKAKOM, 2021. Accessed: Jan. 01, 2023. [Online]. Available: <https://eprints.utdi.ac.id/9323/>
- [10] L. Adi, R. J. Akbar, and W. N. Khotimah, "Platform e-Learning untuk Pembelajaran Pemrograman Web Menggunakan Konsep Progressive Web Apps," *J. Tek. ITS*, vol. 6, no. 2, Art. no. 2, Mar. 2018, doi: 10.12962/j23373539.v6i2.24291.