

# **APLIKASI SETORAN PARKIR BERBASIS PROGRESSIVE WEB APPLICATION DENGAN NOTIFIKASI WHATSAPP GATEWAY**

**Hari Lugis Purwanto**

Prodi Sistem Informasi, Universitas PGRI Kanjuruhan Malang  
hari\_lugis@unikama.ac.id

**Abstract.** Parking deposit management often encounters obstacles ranging from deposit discipline to transaction recording. Field conditions show that parking attendants frequently delay the deposit process, especially during unfavorable weather, while treasurers face time constraints as they work during the same hours as the attendants, making them often unavailable. The limited devices owned by parking attendants also pose a challenge, as their smartphones can only run lightweight applications and are restricted to basic browsing and communication via WhatsApp. These conditions demand a technological solution that is flexible, easy to use, and does not burden the devices used by attendants. The purpose of this study is to implement a Progressive Web Application (PWA) with WhatsApp gateway notifications to support parking deposit reporting. PWA enables web applications to function like mobile applications but with minimal resources and fast access. The results of this study successfully developed a web-based parking deposit application by applying PWA, transforming the web-based deposit program into a mobile-like application that simplifies user access. Users no longer need to type URL addresses through the browser. Furthermore, the implementation of WhatsApp messages or notifications allows real-time communication, while receipt delivery becomes easier and does not consume smartphone storage space since receipts are sent in the form of links.

**Keywords:** *Parking Deposit Application, Progressive Web Application, Whatsapp Gateway, Prototyping*

## **PENDAHULUAN**

Pengelolaan setoran parkir di lingkungan masyarakat sering kali menghadapi berbagai kendala, terutama dalam hal kedisiplinan setoran, pencatatan transaksi, serta komunikasi antara petugas lapangan dan bendahara. Permasalahan ini dialami oleh sebuah perkumpulan pemuda-pemudi yang baru dibentuk untuk menggantikan kepengurusan lama akibat kurangnya kompetensi dalam mengelola area parkir yang berada di salah satu kantor Bank Pemerintah. Pengelolaan parkir tersebut dipercayakan kepada tujuh orang petugas yang bekerja dalam dua *shift*, yaitu *shift* pagi dan *shift* sore, dengan ketentuan nominal setoran harian yang telah ditetapkan.

Dalam praktiknya, proses setoran masih dilakukan secara manual, di mana petugas parkir harus menyerahkan uang hasil parkir secara langsung ke rumah bendahara kemudian bendahara akan membuatkan kuitansi yang di tanda tangani oleh bendahara sebagai penerima setoran dan petugas parkir sebagai penyetor. Keberadaan kuitansi ini sangatlah penting karena berfungsi sebagai perjanjian atau merupakan sebuah indikasi telah terjadinya perjanjian antar pembayar dan orang yang menerima sejumlah uang (Bramantyo et al., 2020). Namun, kondisi di lapangan menunjukkan bahwa petugas parkir sering kali menunda proses setoran, terutama saat kondisi cuaca tidak mendukung seperti hujan. Kendala yang lain adalah bendahara memiliki keterbatasan waktu karena harus bekerja pada jam yang sama dengan jam operasional petugas parkir, sehingga tidak memungkinkan untuk melakukan penagihan langsung. Akibatnya, proses pelaporan keuangan sering mengalami keterlambatan dan berpotensi menimbulkan kesalahpahaman maupun ketidakteraturan dalam pencatatan setoran.

Selain itu, keterbatasan perangkat yang dimiliki oleh para petugas parkir juga menjadi tantangan tersendiri. Dari tujuh petugas, hanya satu petugas yang memiliki *smartphone* dengan spesifikasi memadai, sementara yang lain hanya dapat mengakses aplikasi ringan dan hanya sekedar digunakan untuk komunikasi melalui whatsapp. Kondisi ini menuntut adanya solusi teknologi yang fleksibel, mudah digunakan serta tidak membebani perangkat yang digunakan oleh petugas.

Berdasarkan permasalahan tersebut, dibutuhkan sebuah sistem setoran parkir yang dapat mengelola setoran parkir dengan memperhatikan beberapa kondisi yang terbatas. Penerapan *Progressive Web Application* (PWA) sangat cocok dengan kondisi lapangan. PWA adalah sebuah teknologi baru yang dirancang dan dikembangkan oleh Google pada Juni 2015 untuk mengatasi keterbatasan *browser* seluler dan aplikasi *native* (Herman & Frederick, 2023). Dengan teknologi PWA memungkinkan akses universal, tanpa terbatas oleh lokasi atau perangkat, melalui basis kode tunggal dan teknologi ini dirancang agar pengembang web dapat mengubah situs web yang sudah ada menjadi pengalaman serupa aplikasi mobile tanpa memerlukan penambahan program khusus (Adelin et al., 2024). Selain itu PWA mengukung konsep teknologi terbaru dari *browser* seperti *web apps manifest*, *service worker* dan *app shell* (Hudianti et al., 2023). Dalam analisa pemanfaatan PWA untuk perangkat *mobile* menunjukkan peningkatan signifikan dalam kecepatan dan efisiensi serta meningkatkan kepuasan pengguna tetapi juga dapat memberikan dampak positif pada konversi dan retensi (Hidayat et al., 2023).

Untuk meningkatkan efisiensi dan transparansi komunikasi, sistem juga dilengkapi dengan fitur notifikasi otomatis melalui WhatsApp. Dengan whatsapp memungkinkan pengguna untuk menerima pemberitahuan secara *real time* dan memberikan aksesibilitas dimanapun juga dapat digunakan di *smartphone* maupun laptop, serta notifikasi whatsapp dapat memberikan pesan yang terfokus dan personaliasi yang kuat karena memberikan informasi khusus pada suatu individu dan tidak dapat diketahui pihak lain yang tidak berkepentingan. Disini whatsapp merupakan media sosial yang paling sering digunakan dalam kelangsungan komunikasi organisasi (Yahya & Dirgantara, 2022). Dalam pengembangan sistem ini akan digunakan model *prototyping*. Metode ini mampu meminimalisir kesalahan atau ketidaktepatan analisa kebutuhan system karena segala kekurangan dalam system yang akan di bangun akan terdeteksi pada setiap uji coba *prototype* program untuk kemudian di revisi dan *prototype* di uji kembali (Purwanto, 2021). Model *prototyping* adalah cara cepat untuk mendapatkan informasi tentang kebutuhan pengguna (Wibowo et al., 2024), (Widhiyanti & Atmani, 2021). Dengan demikian dengan adanya sistem ini memungkinkan petugas parkir melakukan konfirmasi setoran secara digital dan mandiri setelah memasukkan uang ke dalam *box* setoran yang tersedia di depan rumah bendahara.

Dengan adanya sistem ini, proses setoran parkir akan menjadi lebih efisien, transparan, dan terdokumentasi secara digital, sekaligus mendukung transformasi digital dalam pengelolaan keuangan komunitas di tingkat masyarakat.

## **METODE PENELITIAN**

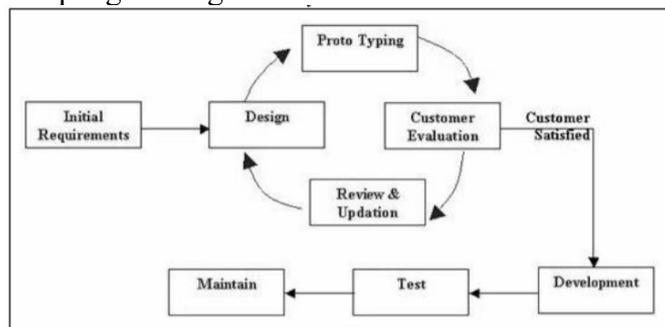
### **Prototyping**

Salah satu kekurangan utama dari metode *Waterfall* adalah munculnya ketidaksamaan persepsi antara analis sistem dengan pengguna. Kondisi ini biasanya terjadi karena pengguna sulit menjelaskan kebutuhan sistem secara rinci. Umumnya, mereka hanya mampu menyampaikan gambaran umum mengenai tujuan perangkat lunak yang diinginkan tanpa mampu merumuskan kebutuhan fungsional secara detail. Sementara itu, pengembang sering kali tidak memiliki kepastian terkait efisiensi algoritma, kemampuan sistem beradaptasi, maupun pola interaksi antara pengguna dan

sistem. Untuk mengatasi permasalahan tersebut, pendekatan *prototyping* menjadi alternatif yang lebih fleksibel. Metode ini menekankan pada pembuatan model awal sistem dalam waktu singkat agar pengguna dapat memberikan masukan secara langsung. Dari umpan balik tersebut, pengembang dapat segera melakukan perbaikan hingga sistem benar-benar sesuai dengan kebutuhan pengguna.

Nugroho (2009) pendekatan *Prototyping* bersifat iteratif dan dapat menutupi kekurangan yang terdapat pada model *Waterfall*, karena:

- Kebutuhan pengguna dapat langsung diwujudkan dalam bentuk sistem yang bisa diuji.
- Sistem dapat disempurnakan secara bertahap berdasarkan umpan balik yang diterima.
- Proses pengembangan melibatkan interaksi dan komunikasi yang intens antara pengguna dan pengembang.



**Gambar 1. Model Prototyping**

### **Progressive Web Application**

*Progressive Web Application* (PWA) merupakan pendekatan pengembangan aplikasi web modern yang menggabungkan kemampuan aplikasi web dan aplikasi *native mobile*. PWA dirancang agar dapat diakses melalui browser tetapi memiliki fitur layaknya aplikasi *native* seperti akses *offline*, *push notification*, dan kemampuan untuk diinstal langsung di layar utama pengguna tanpa melalui toko aplikasi (Fauzan et al., 2022).

Secara teknis, PWA memanfaatkan tiga komponen utama, yaitu *service worker*, *web app manifest*, dan HTTPS. *Service worker* berfungsi sebagai skrip yang berjalan di latar belakang untuk menangani *cache*, sinkronisasi data, serta notifikasi *push* sehingga aplikasi dapat beroperasi meskipun tanpa koneksi internet. *Web app manifest* adalah file konfigurasi berformat JSON yang menentukan nama aplikasi, ikon, warna tema, dan orientasi layar agar *browser* dapat memperlakukan web seperti aplikasi *native*. Sedangkan protokol HTTPS diperlukan untuk menjamin keamanan komunikasi antara *server* dan klien (Saputro & Alfa Aziza, 2025).

Keunggulan utama PWA adalah peningkatan performansi, ketersediaan *offline*, serta instalasi ringan yang tidak membutuhkan ruang besar di perangkat pengguna. Hal ini membuat PWA menjadi solusi efisien untuk organisasi kecil atau komunitas yang ingin menyediakan aplikasi berbasis web tanpa harus mengembangkan dua platform berbeda (web dan mobile). PWA juga mendukung integrasi dengan teknologi lain seperti *microservices* dan *notification gateway*, yang dapat memperluas kemampuan komunikasi antar sistem (Asri et al., 2024). Dengan demikian, PWA dapat menjadi alternatif ideal dalam pengembangan sistem informasi berbasis web yang membutuhkan fleksibilitas tinggi, skalabilitas, dan pengalaman pengguna menyerupai aplikasi *native*.

## **HASIL DAN IMPLEMENTASI**

### **Analisa Kebutuhan Sistem**

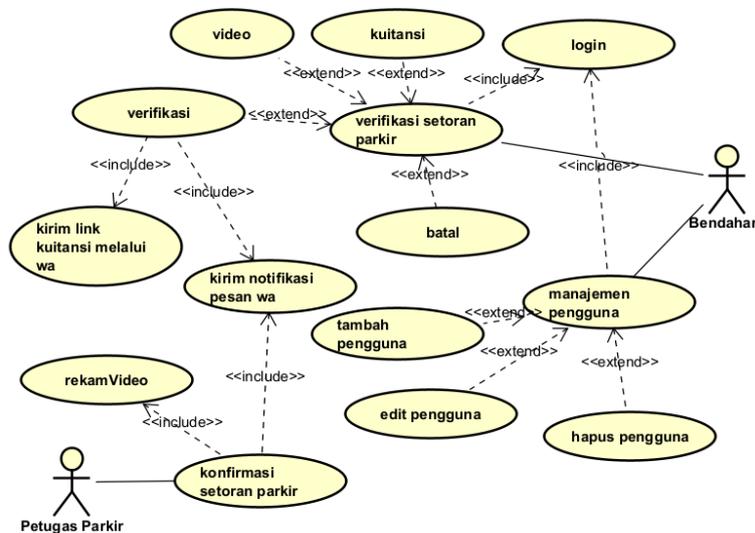
Berdasarkan identifikasi masalah dilapangan, dibutuhkan sistem setoran parkir yang dapat memfasilitasi petugas parkir untuk melakukan setoran secara mandiri dan bendahara bisa melakukan verifikasi setoran dan membuat kuitansi secara digital. Selain itu pada saat petugas parkir melakukan konfirmasi setoran maka perlu ada fitur khusus yang bertugas mengirimkan

informasi melalui whatsapp ke bendahara. Sebaliknya kuitansi juga otomatis dikirimkan melalui whatsapp selepas bendahara melakukan verifikasi setoran yang dilakukan oleh petugas parkir. Oleh karena itu sistem yang akan dibangun harus memiliki spesifikasi kemampuan untuk melakukan pengolahan data sebagai berikut:

1. Data petugas
2. Perekaman video memasukkan setoran ke dalam *box*
3. Konfirmasi setoran parkir oleh petugas parkir
4. Verifikasi setoran parkir oleh bendahara
5. Kuitansi digital
6. Konfirmasi melalui pesan WA secara otomatis pada saat konfirmasi dan verifikasi serta pengiriman *link* kuitansi

**Desain**

Hasil analisa kebutuhan akan digambarkan dalam diagram *use case* seperti pada gambar 2. Terdapat dua aktor yaitu bendahara dan petugas parkir. Petugas parkir bisa mengakses layanan konfirmasi parkir dan sistem otomatis akan mengaktifkan kamera untuk perekaman proses memasukkan uang setoran ke dalam *box* sebelum akses halaman konfirmasi. Pada saat petugas parkir melakukan konfirmasi maka sistem akan otomatis mengirim pemberitahuan berupa pesan whatsapp ke bendahara yang memberitahukan bahwa ada petugas parkir yang telah melakukan setoran mandiri. Kemudian untuk aktor bendahara bisa mengakses layanan manajemen pengguna untuk melakukan manajemen petugas parkir, dan melakukan verifikasi setoran parkir. Ketika bendahara melakukan verifikasi setoran parkir maka sistem akan otomatis mengirimkan link kuitansi melalui pesan whatsapp ke petugas parkir.

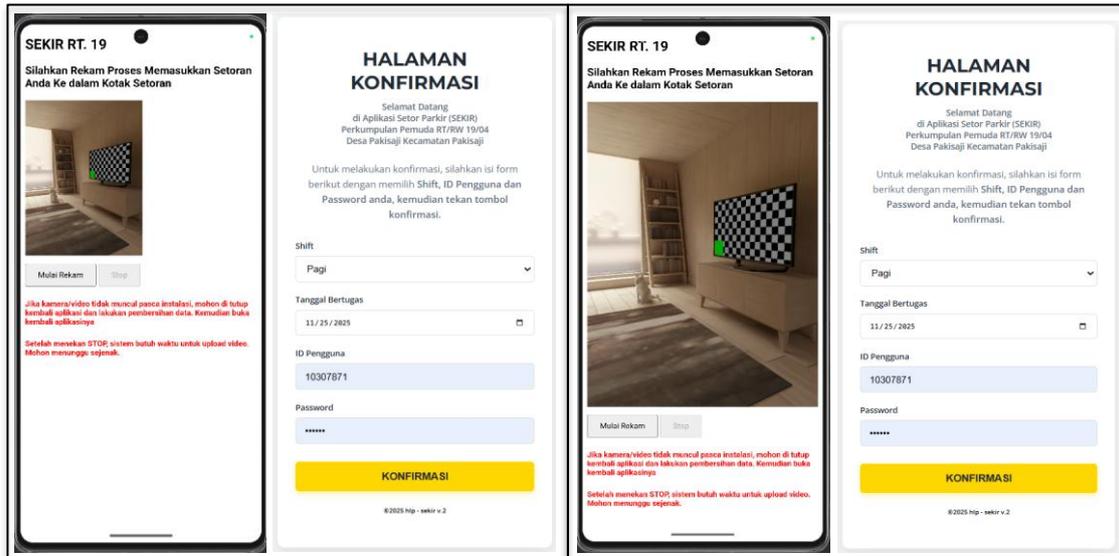


**Gambar 2. Desain Use Case Diagram Sistem Setoran Parkir**

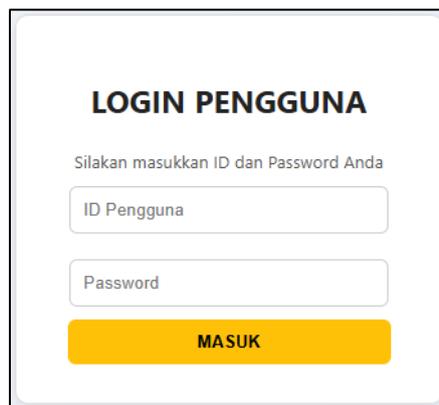
**Prototyping**

*Prototype* dibangun untuk dilakukan uji coba ke calon pengguna untuk mengetahui apakah *prototype* masih ada kekurangan yang perlu ditambah atau tidak. Pada gambar 3 merupakan halaman untuk proses perekaman video yang otomatis akan tampil diawal sebelum *user* mengakses halaman konfirmasi setoran. Halaman konfirmasi setoran akan otomatis terbuka ketika proses perekaman telah dilakukan. Terdapat 4 inputan yang harus dilakukan pada halaman konfirmasi setoran yaitu *shift*, tanggal bertugas, id pengguna dan *password*. Kemudian petugas tinggal menekan tombol konfirmasi.

Pada gambar 4 merupakan gambar layout *login* yang wajib diakses oleh bendahara sebelum mengakses halaman verifikasi setoran parkir dan manajemen pengguna. Proses *login* wajib dipenuhi sebelum mengakses halaman manajemen pengguna dan halaman konfirmasi setoran.

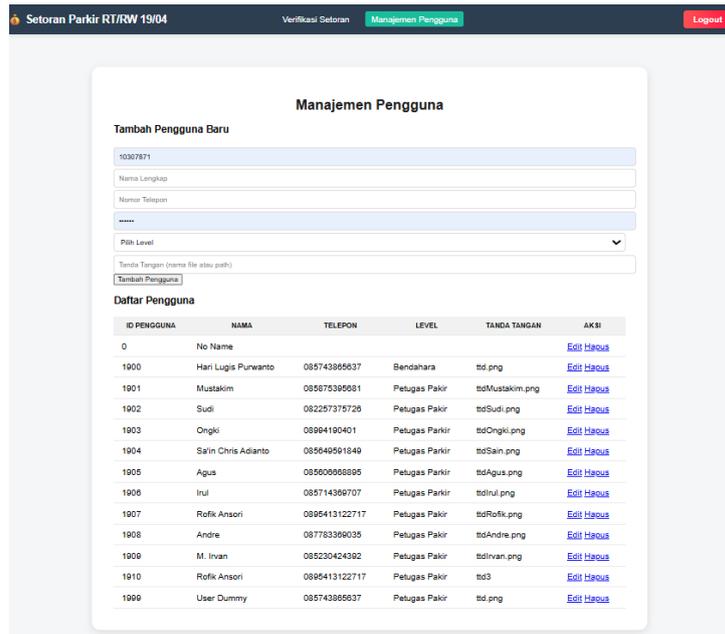


Gambar 3. *Prototype 1* (Kiri) dan *Prototype 2* (Kanan) Halaman Perekaman Video dan Halaman Konfirmasi Setoran

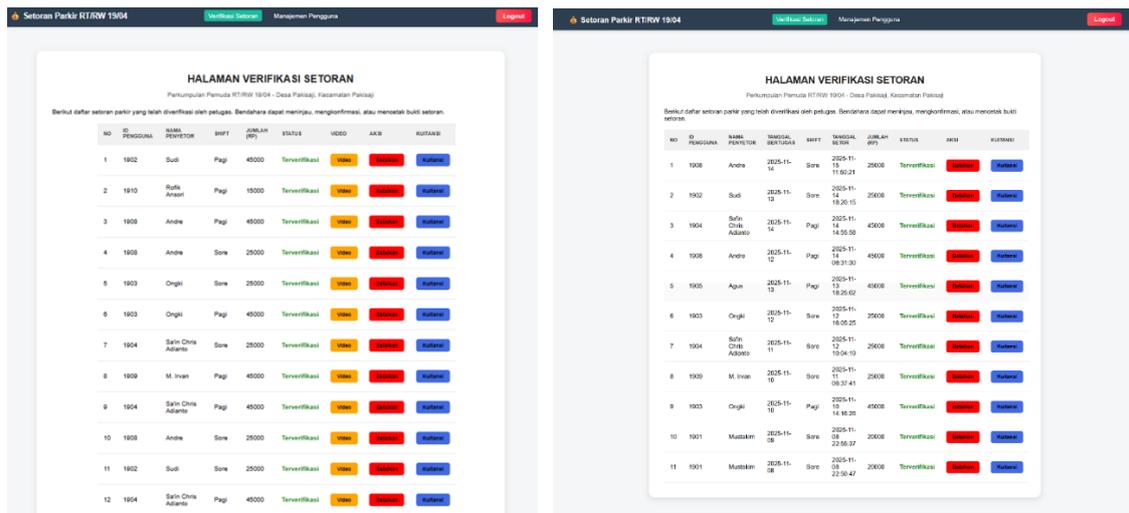


Gambar 4. *Prototype 2* Layout Halaman Login

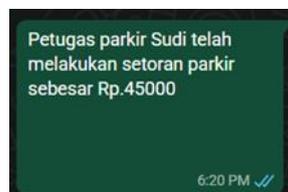
Pada gambar 5 merupakan halaman untuk manajemen pengguna. Dalam halaman ini bendahara bisa melakukan manajemen petugas parkir. Dan gambar ke 6 merupakan halaman verifikasi setoran yang bisa dilakukan oleh bendahara. Ketika bendahara melakukan verifikasi maka otomatis sistem akan mengirimkan pesan ke petugas parkir berupa pemberitahuan bahwa setoran sudah diverifikasi bendahara dan link kuitansi juga dikirimkan melalui pesan whatsapp seperti yang terlihat pada gambar 7 dan 8.



Gambar 5. Prototype 2 Layout Halaman Manajemen Pengguna



Gambar 6 Prototype 1 (kiri) dan Prototype 2 (kanan) Layout Halaman Verifikasi Setoran



Gambar 7. Prototype 2 Pesan Whatsapp ke Bendahara



Gambar 8. Prototype 1 (kiri) dan Prototype 2 (kanan) Pesan Whatsapp ke Petugas Pakir Customer Evaluation

Pengujian dilakukan pada seluruh fitur yang terdapat pada *prototype*. Aktor yang dilibatkan dalam petugas parkir dan bendahara seperti yang terlihat pada tabel 1.

Tabel 1. Data *Feedback User* terhadap *Prototype 1*

Pegguna	Responden	Deskripsi Pengujian	Jumlah Kesesuaian Harapan		Masukan
			Ya	Tidak	
Bendahara	1	Login & Logout	1		
		Halaman Verifikasi Setoran <ul style="list-style-type: none"> <li>- Verifikasi</li> <li>- Pembatalan</li> <li>- Video</li> <li>- Kuitansi</li> </ul>		1	<ul style="list-style-type: none"> <li>- Perlu ditambahkan informasi tanggal bertugas dan tanggal dilakukan setoran.</li> <li>- Tampilan video kurang lebar</li> <li>- Kuitansi tidak boleh di <i>download</i>/di <i>copy</i> untuk melindungi data pengurus. Hanya bisa diakses secara <i>online</i></li> <li>- Link video harus dikirimkan juga melalui pesan WA</li> </ul>
		Halaman Manajemen Pengguna <ul style="list-style-type: none"> <li>- Tambah pengguna</li> <li>- Edit pengguna</li> <li>- Hapus pengguna</li> </ul>	1		
Petugas parkir	7	Konfirmasi Setoran <ul style="list-style-type: none"> <li>- Video perekaman</li> <li>- Konfirmasi setoran</li> </ul>	7		

Sedangkan pada *Feedback* pada *prototype* ke 2 seluruh responden sudah menyatakan tidak ada kekurangan yang di hasilkan sehingga *prototype 2* bisa dilanjutkan pada proses *development*.

### Review dan Updation

Dan hasil dalam evaluasi yang terdapat pada tabel 1 terlihat bahwa ada masukan pada *prototype* yang di uji. Masukan atau *feedback* tersebut adalah perlu adanya penambahan informasi tanggal dilakukan setoran dan tanggal bertugas. Kemudian untuk tampilan video kurang lebar, kuitansi tidak diperkenankan di *download* dan *link* kuitansi juga harus dikirimkan ke pesan WA.

### Development

Setelah tahap *customer evaluation* menunjukkan bahwa pengguna telah merasa puas dan tidak ditemukan lagi kekurangan, maka *prototype* final yang telah disetujui masuk ke tahap penerjemahan ke dalam bahasa pemrograman yang sesuai hingga menjadi aplikasi utuh. Penerapan PWA dilakukan disini. Supaya aplikasi berbasis web dapat diinstall layaknya aplikasi android maka dibutuhkan *script* json untuk memberitahu *browser* bagaimana *web application* seharusnya terlihat dan berperilaku saat di-*install* pada perangkat (*Add to Home Screen*). Tanpa *manifest*, *website* tetap berjalan biasa dan dengan *manifest*, *website* bisa “berubah” jadi aplikasi. Berikut ini *script* yang telah bangun untuk kebutuhan PWA aplikasi setoran parkir.

```
{
  "name": "Setoran Parkir",
  "short_name": "Parkir",
  "start_url": "/videocek.html",
  "scope": "/",
  "display": "standalone",
  "background_color": "#ffffff",
  "theme_color": "#0066ff",
  "description": "Aplikasi rekam video setoran parkir.",
  "icons": [
    { "src": "icons/icon-192.png", "sizes": "192x192", "type": "image/png" },
    { "src": "icons/icon-512.png", "sizes": "512x512", "type": "image/png" }
  ]
}
```

Dengan *script* json tersebut maka *browser* smartphone akan:

1. Mengizinkan instalasi aplikasi dan membuat browser menampilkan *shortcut*/ikon yang bisa dipasang di *homescreen*.
2. Menentukan tampilan saat diluncurkan misalnya *fullscreen/standalone/minimal-ui/browser*.
3. Menentukan *entry point* aplikasi atau *start\_url* (halaman yang dibuka saat *user* klik ikon).
4. Memberi identitas visual seperti ikon, nama, warna tema, *splash screen* (secara tidak langsung).
5. Mendukung pengalaman “*native-like*” seperti konsistensi UI/UX saat pengguna berinteraksi seperti aplikasi *native*.

Kemudian dibutuhkan juga *script* javascript (*service worker*) yang yang dapat berjalan di *background browser*, terpisah dari halaman web. Ia tidak punya akses langsung ke DOM, tapi punya kendali penuh terhadap jaringan, *cache*, dan proses *background*. Dalam PWA, *service worker* adalah *otak* yang membuat *web application* terasa seperti aplikasi *mobile*. Dan berikut ini adalah javascript yang telah dibangun.

```
if ("serviceWorker" in navigator) {
  window.addEventListener("load", () => {
    navigator.serviceWorker
      .register("service-worker.js")
      .then(() => console.log("Service Worker terdaftar"))
      .catch(err => console.log("SW gagal:", err));
  });
}
```

*Service worker* akan menangkap semua *request* halaman (HTML, CSS, JS, gambar) dan bisa mengambil *file* dari internet, atau mengambil *file* dari *cache* jika *offline*, atau menggabungkan keduanya (*hybrid*). Dengan *caching* aplikasi tetap bisa dibuka tanpa internet, halaman loading jadi jauh lebih cepat, dan aplikasi terasa seperti *native*.

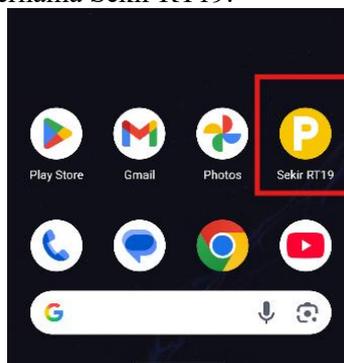
Dan terakhir dibutuhkan lagi javascript untuk yang berfungsi sebagai sebagai *initializer* atau penghubung utama antara aplikasi web dan fitur-fitur PWA sehingga memungkinkan terjadinya fungsi-fungsi sebagai berikut:

- Registrasi *service worker*
- Registrasi *event* instalasi
- Penanganan *prompt* pemasangan aplikasi (*Add to Home Screen*)
- Pengecekan status instalasi
- Notifikasi kepada *user*
- Logika pembantu instalasi PWA

Dan berikut ini adalah javascript yang telah dibangun.

```
if ("serviceWorker" in navigator) {  
  window.addEventListener("load", () => {  
    navigator.serviceWorker  
      .register("service-worker.js")  
      .then(() => console.log("Service Worker terdaftar"))  
      .catch(err => console.log("SW gagal:", err));  
  });  
}
```

Dengan penerapan PWA maka web setoran parkir bisa diinstall di *smartphone* seperti pada gambar 9 dengan icon “P” dan bernama Sekir RT19.



**Gambar 9. Icon Aplikasi Setoran Parkir (Sekir) Setelah diterapkan PWA**

## KESIMPULAN

Berdasarkan pembahasan hasil penelitian diatas didapatkan kesimpulan bahwa program yang dibangun berbasis web dapat menjadi sebuah aplikasi dengan penerapan PWA. Dengan PWA menjadikan program berbasis web seolah menjadi sebuah aplikasi *mobile* sehingga pengguna tidak lagi mengetikkan alamat URL. Dengan penerapan pesan atau notifikasi whatsapp menjadikan proses pemberitahuan terjadi secara *real time* dan pengiriman kuitansi lebih mudah dan tidak memakan *space* penyimpanan *smartphone* petugas karena kuitansi tidak dikirimkan dalam bentuk gambar atau file akan tetapi melalui *link*. Untuk membuka kuitansi petugas cukup membuka aplikasi whatsapp dan membuka link yang telah dikirimkan tanpa membuka aplikasi setoran parkir lagi.

Saran untuk penelitian lanjutan adalah bisa meningkatkan disisi keamanan yaitu akses aplikasi dibatasi pada lokasi tempat *box* setoran parkir berada dengan menentukan posisi *latitude* dan *longitude* area. Kemudian juga bisa dikembangkan lagi dengan autentikasi biometrik sehingga lebih aman dan praktis dibandingkan menginputkan *password* manual.

**DAFTAR PUSTAKA**

- Adelin, N. M., Muhammad, M. A., Wintoro, P. B., & Akhir, K. (2024). Implementasi Progressive Web Apps (PWA) Untuk Meningkatkan Kinerja dan Performa Situs Maritimpreneur. *Jurnal Informatika Dan Teknik Elektro Terapan*, 12(1). <https://doi.org/10.23960/jitet.v12i1.3904>
- Nugroho, A. (2009). *Rekayasa Perangkat Lunak Menggunakan UML dan Java*. Andi Publisher.
- Asri, S. A., Caturbawa, I. G. N. B., Prihatini, P. M., Rasmini, N. W., Nugroho, M. R. A., Rudiastari, E., & Setiawan, W. (2024). Implementation of PWA in Scholarship Application Using Microservices Architecture for Enhancing User Engagement. *JCSTS*, 6. <https://doi.org/10.32996/jcsts>
- Bramantyo, R. Y., Murti, H., Wahyuni, N., & Suwarno. (2020). PENGGUNAAN KUITANSI SEBAGAI ALAT BUKTI TRANSAKSI JUAL BELI (Ditinjau Dari Perspektif Kitab Undang-Undang Hukum Perdata). *Jurnal Ilmu Sosial Dan Ilmu Administrasi Negara*, 4(1), 2620–5149.
- Fauzan, R., Krisnahati, I., Nurwibowo, B. D., & Wibowo, D. A. (2022). A Systematic Literature Review on Progressive Web Application Practice and Challenges. *IPTEK The Journal for Technology and Science*, 33(1), 43. <https://doi.org/10.12962/j20882033.v33i1.13904>
- Herman, & Frederick. (2023). Progressive Web Apps Pengembangan dan Studi Penerimaan pada Mahasiswa Indonesia Menggunakan Scrum dan Model UTAUT. *Jurnal Teknologi Terpadu*, 9.
- Hidayat, R., Hasbi, H. F., Supriani, N., Putri, D. M., Ningrum, S. W., & Farida, M. (2023). Technology and Informatics Insight Journal Rancangan User Interface untuk Meningkatkan Performa Pada E-Commerce Menggunakan Progressive Web Apps (PWA). *Technology and Informatics Insight Journal*, 2. <https://jurnal.universitaspurtrabangsa.ac.id/index.php/tiij>
- Hudianti, E., Mulana, D., & Nugroho, M. A. (2023). Implementasi Progresive Web Apps untuk Sistem Pengelolaan Potensi Desa Wisata Kali Opak Tujuh Bulan. *Journal of Information System Management*, 4.
- Purwanto, H. L. (2021). RANCANG BANGUN SISTEM INFORMASI AKADEMIK BIMBINGAN BELAJAR MENGGUNAKAN MODEL PROTOTYPING. *Jurnal Teknologi, Informasi Dan Industri*, 4.
- Saputro, P. R., & Alfa Aziza, R. F. (2025). PWA AND NON-PWA PERFORMANCE ANALYSIS: CHROME EXTENSION TESTING ON E-COMMERCE PLATFRORM. *JITK (Jurnal Ilmu Pengetahuan Dan Teknologi Komputer)*, 10(4), 909–916. <https://doi.org/10.33480/jitk.v10i4.6238>
- Wibowo, M. A., Purwanto, H. L., & Susanto, G. (2024). RANCANG BANGUN MARKETPLACE PRODUK UMKM DI KABUPATEN NATUNA MENGGUNAKAN MODEL PROTOTYPING. *Journal of Information Technology*, 2(1).
- Widhiyanti, K., & Atmani, A. K. P. (2021). Penerapan Metode Prototyping Dalam Perancangan Interface Sistem Unggah Portofolio Penerimaan Mahasiswa Baru Diploma ISI Yogyakarta. *Teknika*, 10(2), 88–95. <https://doi.org/10.34148/teknika.v10i2.308>
- Yahya, A., & Dirgantara, P. (2022). EFEKTIVITAS PENGGUNAAN WHATSAPP SEBAGAI MEDIA KOMUNIKASI ORGANISASI PROGRAM PTSL DI LINGKUNGAN PEGAWAI DINAS ATR/BPN KOTA PALEMBANG. *MEDIUM*, 10(1), 409–420.