PENGEMBANGAN *FRONT-END* SISTEM *ROVER DRONE* OTONOM BERBASIS *PWA* MENGGUNAKAN *REACT.JS* UNTUK PEMANTAUAN LINGKUNGAN PERKEBUNAN SAWIT

(Skripsi)

Oleh:

MUHAMMAD SALMAN ABDUROHMAN
NPM. 2165031001

JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
2025

ABSTRAK

PENGEMBANGAN *FRONT-END* SISTEM *ROVER DRONE* OTONOM BERBASIS *PWA* MENGGUNAKAN *REACT.JS* UNTUK PEMANTAUAN LINGKUNGAN PERKEBUNAN SAWIT

Oleh

MUHAMMAD SALMAN ABDUROHMAN

Penelitian ini berfokus pada pengembangan antarmuka pengguna *(front-end)* sistem *rover drone* otonom berbasis *Progressive Web App (PWA)* menggunakan *React.js* untuk pemantauan lingkungan perkebunan kelapa sawit. Latar belakang penelitian dilandasi oleh kebutuhan akan sistem pemantauan agronomis yang efisien pada lahan perkebunan berskala luas, yang selama ini masih terkendala oleh keterbatasan konektivitas internet, lambatnya proses pemantauan manual, serta rendahnya efisiensi pengelolaan sumber daya.

Sistem yang dirancang memanfaatkan teknologi *PWA* dengan pendekatan offline-first sehingga mampu tetap berfungsi optimal pada kondisi jaringan yang terbatas atau tidak stabil. *React.js* dipilih sebagai kerangka kerja *front-end* karena kemampuannya dalam membangun antarmuka pengguna yang interaktif, responsif, serta modular, sehingga mendukung integrasi data *real-time* dari sensor *rover drone* secara efisien.

Metode pengembangan menggunakan pendekatan Kanban, meliputi analisis kebutuhan, perancangan, implementasi, pengujian, hingga evaluasi performa sistem. Pengujian dilakukan melalui *Google PageSpeed Insights*, *Blackbox Testing*, serta uji kapabilitas *PWA* untuk memastikan kecepatan, stabilitas, dan kemudahan penggunaan aplikasi.

Hasil penelitian menunjukkan bahwa sistem mampu menampilkan data lingkungan secara *real-time* dengan performa yang responsif, mendukung fungsi *offline*, dan memiliki antarmuka yang ramah pengguna. Kontribusi ilmiah penelitian ini terletak pada integrasi teknologi *PWA* dan *React.js* dalam sistem pertanian digital, yang mendukung pengelolaan perkebunan sawit berbasis data *real-time* secara efisien, adaptif, dan berkelanjutan.

Kata Kunci: React.js, *Progressive Web App*, *Drone* Otonom, Perkebunan Sawit, Pertanian Presisi, Teknologi Pertanian Digital

ABSTRACT

DEVELOPMENT OF A PWA-BASED FRONT-END FOR AN AUTONOMOUS ROVER DRONE SYSTEM USING REACT.JS FOR OIL PALM PLANTATION ENVIRONMENTAL MONITORING

By

MUHAMMAD SALMAN ABDUROHMAN

This research focuses on the development of a user interface (front-end) for an autonomous rover drone system based on Progressive Web App (PWA) technology using React.js for environmental monitoring in oil palm plantations. The study is motivated by the need for an efficient agronomic monitoring system for large-scale plantations, which often face challenges such as limited internet connectivity, time-consuming manual monitoring processes, and low resource management efficiency.

The system was designed using PWA technology with an offline-first approach to ensure optimal functionality even under unstable or limited network conditions. React.js was selected as the front-end framework due to its capability to create interactive, responsive, and modular user interfaces, enabling efficient real-time data integration from rover drone sensors.

The development methodology employed the Kanban approach, covering requirements analysis, design, implementation, testing, and system performance evaluation. Testing involved Google PageSpeed Insights, Blackbox Testing, and PWA capability assessments to ensure application speed, stability, and user-friendliness.

The results indicate that the system can display environmental data in real-time with responsive performance, support offline functionality, and provide a user-friendly interface. The scientific contribution of this research lies in integrating PWA and React.js technologies into digital agriculture systems, enabling efficient, adaptive, and sustainable data-driven management for oil palm plantations.

Keywords: React.js, Progressive Web App, Autonomous Drone, Oil Palm Plantation, Precision Agriculture, Digital Agriculture Technology

PENGEMBANGAN *FRONT-END* SISTEM *ROVER DRONE* OTONOM BERBASIS *PWA* MENGGUNAKAN *REACT.JS* UNTUK PEMANTAUAN LINGKUNGAN PERKEBUNAN SAWIT

Oleh:

MUHAMMAD SALMAN ABDUROHMAN

Skripsi

Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA TEKNIK

Pada

Jurusan Teknik Elektro
Fakultas Teknik Universitas Lampung

FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2025

Judul Skripsi	:	**PENGEMBANGAN *FRONT-END* SISTEM *ROVER DRONE* OTONOM BERBASIS *PWA* MENGGUNAKAN *REACT.JS* UNTUK PEMANTAUAN LINGKUNGAN PERKEBUNAN SAWIT**
Nama Mahasiswa	:	*Muhammad Salman Abdurohman*
Nomor Pokok Mahasiswa	:	2165031001
Program Studi	:	Teknik Elektro
Fakultas	:	Teknik

MENYETUJUI

1. Komisi Pembimbing

Ing Melvi, S.T.,M.T. Ph.D.	**Aryanto, S.T., M.T.**
NIP. 19730118 200003 2001	NIP. 199006212019031011

2. Mengetahui

Ketua Jurusan Teknik Elektro	Ketua Program Studi Teknik Elektro
Herlinawati, S.T., M.T.	**Sumadi, S.T., M.T.**
NIP.197103141999032001	NIP.197311042000031001

MENGESAHKAN

1. Tim Penguji

Ketua : **Ing. Melvi, S.T.,M.T. Ph.D.** .....

Sekretaris : **Aryanto, S.T., M.T.** .....

Penguji : **Yetti Yuniati, S.T., M.T.** .....

2. Dekan Fakultas Teknik

Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc.

NIP. 19750928 200112 1 002

Tanggal Lulus Ujian Skripsi : **10 September 2025**

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama	:	Muhammad Salman Abdurohman
NPM	:	2165031001
Program Studi	:	Teknik Elektro
Jurusan/Fakultas	:	Teknik Elektro/Teknik
Alamat	:	JL. Yos Sudarso KP. KARANG ANYAR gang Yupiter LK.1 RT. 018 Kelurahan Panjang Utara Kecamatan Panjang Bandar Lampung Prov. Lampung 35241

Dengan ini saya menyatakan bahwa dalam skripsi ini yang berjudul "Pengembangan Front-End Sistem Rover Drone Otonom Berbasis Pwa Menggunakan React.Js Untuk Pemantauan Lingkungan Perkebunan Sawit" tidak terdapat karya yang pernah dilakukan oleh orang lain dan sepanjang pengetahuan saya tidak terdapat atas diterbitkannya oleh orang lain, kecuali secara tertulis diacu dalam naskah ini sebagaimana yang disebutkan dalam daftar pustaka. Selain itu, saya menyatakan pula bahwa skripsi ini dibuat oleh saya sendiri.

Apabila pernyataan saya tidak benar, maka saya bersedia dikenai sanksi akademik sesuai dengan hukum yang berlaku.

Bandar Lampung, 21 November 2025

Muhammad Salman Abdurohman

NPM. 2165031001

RIWAYAT HIDUP

Penulis dilahirkan di Bandar Lampung pada tanggal 21 Februari 2003, sebagai anak kesatu dari tiga bersaudara, dari Bapak Pontas Hasan Basri dan Ibu Tati Yuliyanti. Pendidikan Sekolah Dasar diselesaikan di MIN 5 Sukarame, Bandar Lampung pada tahun 2015, Sekolah Menengah Pertama di MTsN 2 Bandar Lampung diselesaikan pada tahun 2018, dan Sekolah Menengah Atas di MAN 2 Bandar Lampung dengan jurusan IPA yang diselesaikan pada tahun 2021. Tahun 2021, penulis terdaftar sebagai mahasiswa Jurusan Teknik Elektro Universitas Lampung melalui jalur Prestasi khusus (Hafizh). Selama menjadi mahasiswa penulis aktif mengikuti Organisasi Himpunan Mahasiswa Teknik Elektro (Himatro) Universitas Lampung sebagai anggota Divisi Kerohanian tahun 2022 hingga 2023. Penulis juga mengikuti lomba PKM-K dan lolos sampai tahap wawancara pasca pendanaan di tahun 2022. Pada tahun 2023 penulis menjadi ketua departemen Kajian di FOSSI FT Universitas Lampung selama priode bulan Januari 2023 sampai bulan Desember 2023. Penulis juga mengikuti Lomba Tartil Qur'an yang diadakan oleh UNSRI, FKIP UNILA, Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi (Kemendikbudristek) dan beberapa tempat lainnya Penulis mengambil konsentrasi Telekomunikasi dan Teknologi Informasi dan secara aktif mengikuti kegiatan akademik di konsentrasi tersebut. Beberapa kegiatan akademik pada lingkup tersebut yang diikuti oleh penulis di antaranya mengikuti summer course di ITB, MBKM riset microclimate di Wilayah Hutan Wisata Mangrove Petengoran, studi independent di Dicoding dan magang independen yang dilaksanakan di kantor PGNCOM.

PERSEMBAHAN

Dengan segala puji syukur bagi Allah atas berkah rahmat dan karunia-Nya, Kupersembahkan karya ini dengan rasa syukur, hormat dan kasih sayang kepada:

“Kepada Kedua Orang Tuaku,
Bapakku Hasan Basri dan Ibuku Tati Yuliyanti”.

Atas segala doa, kasih sayang, kepercayaan, dan pengorbanan yang tiada henti. Terima kasih telah menjadi sumber kekuatan dan perlindungan dalam setiap langkah perjalanan ini. Berkat kalian, penulis dapat menyelesaikan skripsi ini dengan baik.

“Kepada Diri Sendiri Muhammad Salman Abdurohman ”.

Atas segala usaha, ketekunan, dan semangat yang tak pernah padam, serta atas keteguhan untuk terus bertahan dan melangkah. Skripsi ini tidak sempurna tapi cukup untuk membuat penulis wisuda dan mendapatkan gelar S.T. Semoga ini menjadi awal dari perjalanan yang lebih baik, dan kiranya segala langkah ke depan selalu disertai oleh Allah

MOTTO

وَمَنْ يَّتَّقِ اللّٰهَ يَجْعَلْ لَّهٗ مِنْ اَمْرِهٖ يُسْرًا ٤

Artinya: “Siapa yang bertakwa kepada Allah, niscaya Dia menjadikan kemudahan baginya dalam urusannya”. (Q.S At-Talaq: 4)

مَنْ نَفَّسَ عَنْ مُؤْمِنٍ كُرْبَةً مِنْ كُرَبِ الدُّنْيَا نَفَّسَ اللَّهُ عَنْهُ كُرْبَةً مِنْ كُرَبِ يَوْمِ الْقِيَامَةِ وَمَنْ يَسَّرَ عَلَى مُعْسِرٍ يَسَّرَ اللَّهُ عَلَيْهِ فِى الدُّنْيَا وَالآخِرَةِ

Artinya: " Barang siapa melapangkan kesulitan seorang mukmin dari kesulitan di dunia, maka Allah SWT akan memudahkan kesulitan-kesulitannya pada hari kiamat. Siapa yang memudahkan orang yang sedang kesulitan niscaya Allah mudahkan baginya di dunia dan akhirat." (H.R. Muslim).

"Perjuangan yang menggenggam tangan Allah akan selalu berakhir pada kemenangan, kebaikan, dan keindahan." (Muhammad Salman Abdurohman)

SANWACANA

Segala puji dan syukur penulis panjatkan kepada Allah yang maha esa sehingga dengan izin-Nya penulis dapat menyelesaikan skripsi yang berjudul "Pengembangan *Front-End* Sistem *Rover Drone* Otonom Berbasis *Pwa* Menggunakan React.*J*s Untuk Pemantauan Lingkungan Perkebunan Sawit" dengan baik. Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik di Jurusan Teknik Elektro, Fakultas Teknik, Universitas Lampung. Dalam proses penyusunannya, penulis mendapatkan dukungan dari berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada semua pihak yang telah membantu, khususnya kepada :

1. Ibu Prof. Dr. Ir. Lusmeilia Afriani, D.E.A., I.P.M., selaku Rektor Universitas Lampung.
2. Bapak Dr. Eng. Helmy Fitriawan, S.T., M.Sc. selaku Dekan Fakultas Teknik Universitas Lampung.
3. Ibu Herlinawati, S.T., M.T. selaku Ketua Jurusan Teknik Elektro Fakultas Teknik Universitas Lampung.
4. Bapak Sumadi, S.T., M.T. selaku Kepala Program Studi Teknik Elektro Universitas Lampung.
5. Aryanto, S.T., M.T. selaku dosen pembimbing akademik dan dosen pembimbing pendamping dari skripsi yang telah memberikan arahan dan dukungan kepada penulis.

6. Bapak Ing Melvi, S.T., M.T., Ph.D. selaku Dosen Pembimbing Utama atas segala waktu yang telah diluangkan untuk memberikan bimbingan, nasihat, arahan dan juga motivasi yang membangun kepada penulis.

7. Ibu Yetti Yuniati, S.T., M.T.selaku Dosen Penguji atas segala waktu yang telah diluangkan untuk memberikan bimbingan, nasihat, arahan dan juga motivasi yang membangun kepada penulis.
8. Segenap dosen di Jurusan Teknik Elektro yang telah memberikan ilmu yang bermanfaat kepada penulis selama menempuh pendidikan di perkuliahan.
9. Penulis menyampaikan rasa terima kasih yang sedalam-dalamnya kepada kedua orang tua tercinta, Bapak Hasan Basri dan Ibu Tati Yuliyanti, atas segala cinta, doa, dukungan, dan pengorbanan yang tiada henti sejak awal hingga detik ini. Terima kasih atas ketulusan dalam membesarkan, membimbing, dan mendampingi penulis dengan kasih yang tidak ternilai. Tanpa kekuatan doa, kerja keras, dan semangat yang terus Bapak dan Ibu tanamkan, pencapaian ini tidak akan pernah menjadi nyata. Semoga Allah membalas setiap kebaikan dan jerih payah Bapak dan Ibu dengan berkat yang berlimpah.
10. Ucapan terima kasih kepada keluarga besarku, yang selalu menjadi sumber inspirasi dan motivasi serta memberikan dukungan tanpa henti sepanjang penulisan skripsi ini. Maaf atas sikapku yang sering kali keras kepadamu. Namun, ketahuilah bahwa rasa sayangku padamu sangat besar, dan semua itu lahir dari keinginanku agar dirimu bisa menjadi pribadi yang lebih baik dariku. Terima kasih telah selalu bersabar, memahami, dan tetap mendukungku.
11. Seluruh teman-teman Telekomunikasi dan Teknologi Informasi 2021 yang telah memberikan dukungan, semangat dan motivasi kepada penulis untuk menyelesaikan skripsi ini.
12. Keluarga Besar Himatro Unila yang telah memberikan pengalaman tak terlupakan kepada penulis.
13. Keluarga besar FOSSI FT 2021 yang telah pengalman keagamaan yang selalu akan ku ingat.
14. Kepada seluruh teman teman komunitas Quran baik yang di unila maupun diluar unila, baik yang selalu offline maupun yang terkadang offline. Kalian

15. adalam komunitas yang selalu memberi banyak dukungan dalam keistiqomahan Bersama Al-Quran disaat dunia ini menyibukkanku.
16. Kepada teman teman yang sering berkumpul dan bermalam Bersama di masjid AL-Fatih (Masjid Fakultas Teknik UNILA) yang selaalu Bersama dalam perjuangan menmpuh Pendidikan selama ini

Akhir kata, penulis menyadari bahwa skripsi ini masih jauh dari kata sempurna, maka dari itu penulis mengharapkan kritik dan saran yang membangun dari berbagai pihak. Semoga skripsi ini dapat membantu dan memberikan manfaat bagi penulis dan para pembaca.

Bandar Lampung, 19 November 2025

Muhammad Salman Abdurohman
NPM. 2165031001

DAFTAR ISI

Halaman

DAFTAR GAMBAR

Halaman

DAFTAR TABEL

Halaman

I. PENDAHULUAN

1.1 Latar Belakang

Indonesia merupakan salah satu negara produsen utama minyak kelapa sawit di dunia, dengan kontribusi lebih dari 50% terhadap total produksi global. Komoditas ini memiliki peran strategis dalam perekonomian nasional, khususnya pada sektor pertanian, dengan memberikan sumbangan sekitar 35% terhadap Produk Domestik Bruto (PDB) dan mencapai 12% dari total nilai ekspor nonmigas [1]. Namun demikian, pengelolaan perkebunan kelapa sawit di Indonesia masih sangat bergantung pada metode manual, sehingga efisiensi dalam penggunaan waktu, tenaga, dan biaya operasional masih rendah. Proses pemantauan kondisi tanaman, analisis kesehatan tanah, serta deteksi dini terhadap hama dan penyakit sering kali terlambat dilakukan, yang berpotensi menimbulkan kerugian signifikan [2].

Sebagai solusi, penerapan teknologi *drone rover* menawarkan pendekatan yang lebih efisien dan adaptif. Integrasi sensor dan kamera multispektral memungkinkan pengambilan data secara *real-time* terkait kelembapan tanah, suhu udara, serta kondisi vegetasi secara akurat. Data tersebut membantu pengambilan keputusan yang cepat dan tepat untuk intervensi teknis di lapangan [3]. Mobilitas tinggi *drone rover* juga memudahkan akses ke area yang sulit dijangkau, seperti lahan terpencil atau berkontur ekstrem [4].

Data yang dikumpulkan dapat dimanfaatkan secara optimal, diperlukan sistem antarmuka *web* yang mampu mengelola, menyimpan, dan menampilkan informasi secara efisien. Sistem ini berfungsi memantau distribusi dan

penggunaan sumber daya seperti air dan pupuk, mengurangi pemborosan, serta mendukung upaya pelestarian lingkungan. Data historis yang tersimpan dapat digunakan untuk membangun model prediktif yang mendukung pengambilan keputusan berbasis data jangka panjang [5].

Implementasi *Progressive Web App (PWA)* berbasis *React.js* merupakan landasan utama dalam pengembangan sistem pemantauan digital ini. Teknologi tersebut memungkinkan antarmuka pengguna untuk tetap responsif di berbagai perangkat serta dapat beroperasi dalam kondisi jaringan terbatas melalui pendekatan *offline-first*. Pendekatan ini berkontribusi pada peningkatan efisiensi operasional di lapangan sekaligus memperkuat sistem berbasis data yang berkelanjutan dan inklusif [6].

Tren model penyewaan perangkat *drone rover* pada sektor agrikultur memberikan kemudahan akses teknologi bagi petani. Skema penyewaan ini memungkinkan petani memanfaatkan teknologi canggih tanpa perlu melakukan investasi besar pada perangkat keras. Penggunaan *drone* dapat disesuaikan dengan kebutuhan agronomis, seperti pada masa tanam atau periode pemantauan intensif, sehingga pemanfaatan teknologi menjadi lebih optimal dan efisien.

Penyedia layanan umumnya juga menyediakan pelatihan teknis kepada operator guna memastikan perangkat dapat dioperasikan secara optimal. Data yang diperoleh tidak hanya digunakan untuk pemantauan visual, melainkan juga menjadi dasar dalam pengambilan keputusan agronomi terkait penggunaan pupuk, irigasi, dan pestisida. Pendekatan ini selaras dengan prinsip pertanian presisi dan keberlanjutan. Beberapa penyedia layanan menawarkan analitik lanjutan untuk mengolah data mentah menjadi informasi strategis mengenai kondisi lahan dan tanaman.

Sektor kelapa sawit memiliki potensi signifikan untuk dioptimalkan melalui penerapan teknologi digital. Pengembangan sistem antarmuka *web* berbasis *React.js* dan *Progressive Web Application (PWA)* merupakan langkah krusial dalam mendukung digitalisasi perkebunan sawit yang efisien, adaptif, dan berbasis data.

Penelitian ini bertujuan merancang dan mengimplementasikan antarmuka pengguna *(front-end)* yang mampu mendukung operasional *drone rover* secara efektif dalam sistem pemantauan perkebunan sawit. Sistem yang dikembangkan diharapkan dapat meningkatkan efisiensi pertanian modern, memperkuat transformasi digital di sektor agrikultur, serta mendukung pengambilan keputusan berbasis data secara *real-time.*

1.2 Identifikasi Masalah

Berdasarkan uraian latar belakang, beberapa permasalahan utama yang perlu diselesaikan dalam pengembangan antarmuka pengguna *(front-end)* sistem pemantauan dan pemeliharaan perkebunan kelapa sawit berbasis *Progressive Web App (PWA)* menggunakan *React.js* dengan dukungan data dari *drone rover* adalah sebagai berikut:

1. Pemrosesan dan visualisasi data *real-time* (intensitas cahaya, suhu, dan kelembapan tanah) membutuhkan antarmuka yang cepat, efisien, dan stabil [7].
2. Keterbatasan konektivitas internet di wilayah perkebunan menyebabkan gangguan sinkronisasi data, sehingga sistem harus tetap menyediakan akses meskipun dalam kondisi *offline* [7].
3. Diperlukan visualisasi data interaktif dan intuitif agar pengguna mudah memahami kondisi agronomis secara menyeluruh [8].
4. Pemantauan multi-titik berskala besar masih menjadi tantangan dalam penerapan fitur *offline-first* pada *PWA* [9].
5. Antarmuka harus mudah digunakan oleh pengguna tanpa latar belakang teknis, terutama dalam skema penyewaan perangkat *rover* [10].

Penyelesaian masalah di atas menjadi kunci untuk membangun sistem pemantauan pertanian digital yang tangguh dan adaptif.

1.3 Rumusan Masalah

Berdasarkan uraian pada bagian latar belakang dan identifikasi masalah, maka rumusan masalah dalam penelitian ini dapat dirumuskan sebagai berikut:

1. Bagaimana mengoptimalkan kinerja antarmuka pengguna agar mampu memproses serta menampilkan data secara *real-time* tanpa keterlambatan?
2. Bagaimana mengimplementasikan pendekatan *offline-first* pada arsitektur *Progressive Web App (PWA)* sehingga aplikasi tetap dapat berfungsi secara optimal di wilayah dengan keterbatasan konektivitas internet?
3. Bagaimana merancang visualisasi data yang interaktif, mudah dipahami, dan informatif untuk mendukung pengambilan keputusan cepat dan akurat oleh pengelola perkebunan?

1.4 Tujuan Penelitian

Tujuan penelitian ini sebagai berikut:

1. Mengembangkan antarmuka pengguna yang mampu menampilkan data secara *real-time* dengan efisien dan responsif, sehingga pemantauan melalui *drone rover* dapat berjalan tanpa jeda (*lag*).
2. Menerapkan pendekatan *offline-first* pada *Progressive Web App (PWA)* agar sistem tetap dapat diakses secara optimal meskipun jaringan internet tidak stabil.
3. Merancang sistem visualisasi data yang interaktif dan informatif untuk membantu pengelola perkebunan dalam menilai kondisi lahan serta mengambil keputusan dengan cepat dan akurat.

1.5 Manfaat Penelitian

Penelitian ini diharapkan memberikan manfaat sebagai berikut:

1. Meningkatkan efisiensi operasional pemantauan dan pemeliharaan tanaman melalui pemanfaatan data *real-time*, sekaligus menekan biaya

   operas ional dan meningkatkan produktivitas lahan.

2. Menjadi pijakan awal bagi pengembangan integrasi teknologi *drone* dan *PWA* dalam sistem agrikultur digital yang dapat diadaptasi pada subsektor pertanian lainnya.
3. Memberikan kontribusi ilmiah sebagai referensi pengembangan sistem antarmuka pengguna berbasis data dan teknologi pertanian presisi.

1.6 Batasan Masalah

Untuk menjaga fokus penelitian, batasan masalah yang ditetapkan adalah:

1. Sebagian besar data merupakan data simulasi (*dummy*), dengan satu set data nyata digunakan untuk validasi awal. Pendekatan ini memastikan sistem dapat diuji meskipun data lapangan lengkap belum tersedia.
2. Visualisasi dibatasi pada parameter lingkungan dasar, yaitu intensitas cahaya, suhu udara, dan kelembapan tanah yang diperoleh dari *drone rover*. Fokus diarahkan pada efisiensi pemrosesan dan penyajian data.
3. Pengujian dilakukan pada perkebunan berskala terbatas untuk memastikan kestabilan dan kinerja sistem dalam kondisi terkendali.
4. Fitur *offline-first* difokuskan pada skenario gangguan konektivitas jangka pendek. Sistem tetap berfungsi meskipun jaringan tidak stabil, dengan asumsi koneksi kembali dalam waktu singkat. Penelitian ini tidak mencakup skenario tanpa jaringan sama sekali.

Batasan ini memengaruhi desain sistem, strategi pengujian performa, dan pendekatan pengembangan antarmuka pengguna yang akan dijelaskan pada Bab IV. Dengan ruang lingkup terukur dan realistis, penelitian ini diharapkan menghasilkan sistem yang sesuai dengan kebutuhan teknis di lapangan.

1.7 Sistematika Penulisan

Sistematika penulisan skripsi ini disusun secara terstruktur dalam lima bab utama, yang masing-masing memiliki cakupan sebagai berikut:

BAB I – PENDAHULUAN

Menjelaskan latar belakang, identifikasi masalah, rumusan masalah, batasan, tujuan, manfaat, dan sistematika penulisan.

BAB II – TINJAUAN PUSTAKA

Membahas teori pendukung seperti konsep *drone rover*, *Progressive Web App (PWA)*, dan *React.js*, serta studi terdahulu yang relevan sebagai dasar pengembangan sistem.

BAB III – METODOLOGI PENELITIAN

Menjelaskan metode penelitian menggunakan pendekatan *Kanban* yang fleksibel dan bertahap, meliputi analisis kebutuhan, perancangan, pembuatan, pengujian, serta evaluasi sistem.

BAB IV – HASIL DAN PEMBAHASAN

Menyajikan hasil pengembangan *front-end* serta hasil pengujian kinerja sistem di lapangan, dengan fokus pada efektivitas dan efisiensi operasional.

BAB V – KESIMPULAN DAN SARAN

Menyimpulkan hasil penelitian, menjawab rumusan masalah, dan memberikan saran untuk pengembangan sistem lebih lanjut agar kinerja dan fungsionalitas meningkat.

II. TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Bagian ini mengkaji sejumlah penelitian terdahulu yang relevan dengan pengembangan *front-end* untuk sistem pemantauan perkebunan berbasis *Progressive Web App (PWA)* menggunakan *React.js*. Kajian ini bertujuan memetakan pendekatan, teknologi, serta temuan utama dari berbagai studi terdahulu. Selain itu, kajian ini dilakukan untuk mengidentifikasi celah penelitian yang dapat diisi melalui pengembangan sistem ini.

Anderson dan Gaston (2019) [11], dalam penelitian berjudul *"Lightweight Unmanned Aerial Vehicles Will Revolutionize Spatial Ecology"*, menyoroti potensi pesawat tanpa awak (*Unmanned Aerial Vehicles/UAV*) dalam mendukung kegiatan pemantauan ekologis. *UAV* berbiaya rendah memungkinkan pengumpulan data berulang mengenai dinamika vegetasi dan kondisi lingkungan secara efisien. Dalam konteks pertanian, teknologi ini digunakan untuk mengidentifikasi stres air dan kesehatan tanaman, sehingga mendukung sistem pemantauan presisi berbasis data spasial.

Shamrat et al. (2020) [12], melalui penelitian berjudul *"A Web-Based Application for Agriculture: Smart Farming System"*, mengembangkan aplikasi berbasis *web* untuk mendukung petani di Bangladesh. Sistem tersebut memberikan akses terhadap informasi penyakit tanaman, konsultasi dengan ahli, serta forum berbagi pengetahuan. Aplikasi ini dibangun menggunakan *HTML*, *CSS*, dan *PHP*. Hasil penelitian menunjukkan bahwa pemanfaatan teknologi digital mampu meningkatkan akses informasi serta efisiensi operasional di sektor pertanian. Studi

ini juga menegaskan pentingnya antarmuka berbasis *web* sebagai jembatan antara petani dengan data pertanian yang relevan.

Lopez Contreras et al. (2024) [13] mengembangkan antarmuka *web* untuk teleoperasi *rover* pada lingkungan ekstrem. Sistem tersebut memungkinkan pengguna mengendalikan *rover* dari jarak jauh dengan mengirim perintah sekaligus menampilkan data sensor secara *real-time* melalui peramban. Antarmuka ini dirancang bagi pengguna yang tidak memiliki akses langsung ke perangkat keras *rover*. Pemanfaatan antarmuka berbasis web menunjukkan bahwa *front-end* berperan penting sebagai penghubung antara operator dan *rover*. Hal ini relevan dengan pengembangan sistem berbasis *React.js* dan *Progressive Web App (PWA)* yang mengintegrasikan kendali *rover* serta visualisasi telemetri secara daring.

Penerapan nyata *Open MCT* dari NASA juga dapat dijadikan rujukan karena mampu memvisualisasikan serta mengontrol *rover* dan data misi secara berbasis web. Prototipe *dashboard React.js + ROSBridge* dari beberapa proyek universitas pun menunjukkan integrasi *front-end* modern dengan *ROS* untuk sistem *rover* agrikultur.

Ketiga penelitian tersebut menekankan pentingnya integrasi antara teknologi drone/*rover*, aplikasi *web*, dan pendekatan berbasis data untuk meningkatkan efisiensi sistem pemantauan lingkungan dan pertanian. Namun, sebagian besar studi masih terbatas pada pengujian sistem secara terpisah dan belum menitikberatkan pada pengembangan antarmuka pengguna yang responsif, adaptif, serta mendukung skenario dengan konektivitas terbatas.

Berdasarkan tinjauan tersebut, penelitian ini diarahkan untuk mengisi celah yang ada. Fokus penelitian adalah pada pengembangan antarmuka pengguna berbasis *React.js* dalam bentuk *Progressive Web App (PWA)* yang mampu menampilkan visualisasi data secara *real-time*, berfungsi pada mode *offline*, serta terintegrasi dengan sistem pemantauan *drone rover* guna mendukung operasional perkebunan kelapa sawit secara efisien dan berkelanjutan.

2.2 *Drone Rover* dalam Aplikasi Pertanian

Dalam sistem pertanian presisi, teknologi *drone rover* telah berkembang menjadi salah satu solusi penting dalam kegiatan pemantauan dan pengelolaan lahan. Kombinasi antara *drone* udara dan kendaraan darat otonom memungkinkan proses akuisisi data lingkungan dilakukan secara menyeluruh, baik secara vertikal maupun horizontal.

Penerapan *drone rover* pada sektor perkebunan kelapa sawit memberikan peluang besar untuk melakukan pengumpulan data agronomis secara efisien dan berkelanjutan, terutama pada lahan berskala luas yang sulit dijangkau dengan metode konvensional. Teknologi ini memiliki kemampuan mobilitas tinggi serta integrasi dengan berbagai sensor modern. Melalui integrasi tersebut, sistem *drone rover* dapat memantau berbagai parameter lingkungan secara akurat untuk mendukung pengambilan keputusan berbasis data.

Penjelasan mengenai komponen dan fungsi utama *drone rover* dalam aplikasi pertanian disajikan pada tabel berikut.

Tabel 2.1 Komponen Drone Rover [10]

No.	Aspek	Deskripsi
1.	*Drone Rover*	Kombinasi antara drone udara dan kendaraan darat otonom, inovasi penting dalam aplikasi pertanian modern.
2.	*Drone* Udara	Pesawat tanpa awak yang dapat terbang dan mengumpulkan data dari ketinggian.
3.	Kendaraan Darat Otonom	Kendaraan yang dapat bergerak secara mandiri tanpa pengemudi, menggunakan teknologi seperti sensor dan algoritma navigasi.

Tabel 2.1 menjelaskan bahwa *drone rover* merupakan kombinasi antara *drone* udara (*Unmanned Aerial Vehicle/UAV*) dan kendaraan darat otonom yang bekerja secara terintegrasi untuk mendukung penerapan pertanian presisi, khususnya dalam pengelolaan perkebunan kelapa sawit. *Drone* udara berfungsi mengumpulkan data spasial dari ketinggian, sedangkan kendaraan darat dilengkapi sensor dan algoritma

navigasi untuk menjelajah lahan secara mandiri tanpa intervensi langsung manusia [10].

Dalam praktiknya, *drone rover* dilengkapi dengan sensor lingkungan dan kamera multispektral untuk mengukur parameter seperti kelembapan tanah, suhu udara, dan intensitas cahaya. Selain itu, perangkat ini juga mampu menangkap citra vegetasi pada berbagai spektrum. Teknologi tersebut memungkinkan analisis kondisi tanaman secara *real-time*, termasuk deteksi stres tanaman, pemetaan kesehatan vegetasi, serta identifikasi dini terhadap hama dan penyakit. Dengan demikian, sistem ini mendukung pengambilan keputusan yang cepat, akurat, dan berbasis data.

Antarmuka pengguna berbasis *web* dikembangkan menggunakan *React.js*, yaitu pustaka *JavaScript* modern yang memungkinkan penampilan data secara dinamis dan *real-time* tanpa memerlukan pemuatan ulang halaman. Pendekatan ini meningkatkan efisiensi visualisasi data bagi pengelola perkebunan. Sementara itu, penerapan *Progressive Web App (PWA)* memberikan keunggulan berupa akses lintas perangkat dan dukungan *offline-first* melalui *Service Worker*, yang memungkinkan penyimpanan data secara lokal serta sinkronisasi otomatis ketika koneksi internet kembali tersedia.

Dengan kombinasi teknologi tersebut, sistem *drone rover* mampu memberikan pemantauan agronomis yang efisien, adaptif, dan andal, bahkan di wilayah dengan keterbatasan jaringan.

2.3. Tantangan dan Kebutuhan Teknologi Perkebunan Kelapa Sawit

Perkebunan kelapa sawit merupakan salah satu sektor strategis dalam perekonomian Indonesia karena memberikan kontribusi signifikan terhadap Produk Domestik Bruto (PDB) nasional dan ekspor nonmigas. Luas lahan yang melebihi 14 juta hektare mencerminkan besarnya skala industri ini. Kondisi tersebut

sekaligus menunjukkan tantangan besar dalam hal efisiensi pemantauan dan keberlanjutan operasional.

Gambaran yang lebih komprehensif mengenai peran dan kompleksitas sektor ini, Tabel 2.2 berikut menyajikan ringkasan tantangan utama serta kebutuhan teknologi yang relevan dalam pengelolaan perkebunan kelapa sawit.

Tabel 2.2 Informasi Perkebunan Kelapa Sawit [16]

No.	Aspek	Deskripsi
1.	Perkebunan Kelapa Sawit	Sektor penting bagi ekonomi Indonesia, memberikan kontribusi besar terhadap PDB dan ekspor non-migas.
2	Luas Lahan	Lebih dari 14 juta hektar, mencakup area yang luas dan beragam.
3	Tantangan	Menghadapi tantangan dalam pemantauan dan pemeliharaan lahan, terutama di daerah terpencil.
4	Pemantauan Manual	Sering kali tidak efisien dalam mendeteksi masalah di wilayah yang luas.

Tabel 2.2 menunjukkan bahwa perkebunan kelapa sawit memiliki peran penting dalam perekonomian Indonesia. Namun, luas lahan yang sangat besar menimbulkan tantangan signifikan dalam kegiatan pemantauan dan pemeliharaan, terutama di daerah yang sulit dijangkau secara manual.

Pemantauan konvensional pada lahan berskala besar sering kali tidak efektif dalam mendeteksi kelembapan tanah, kondisi tanaman, maupun potensi serangan hama atau penyakit secara cepat dan akurat. Keterlambatan deteksi dapat menyebabkan kerugian ekonomi karena lambatnya tindakan korektif di lapangan.

Pengembangan sistem pemantauan berbasis *Progressive Web App (PWA)* menggunakan React.js sebagai *framework* frontend dilakukan untuk mengatasi kendala tersebut [16].Teknologi *Progressive Web App* (*PWA)* menggabungkan keunggulan aplikasi web dan mobile dengan sifat ringan, responsif, serta mendukung konsep *offline-first* melalui *Service Worker* yang memungkinkan

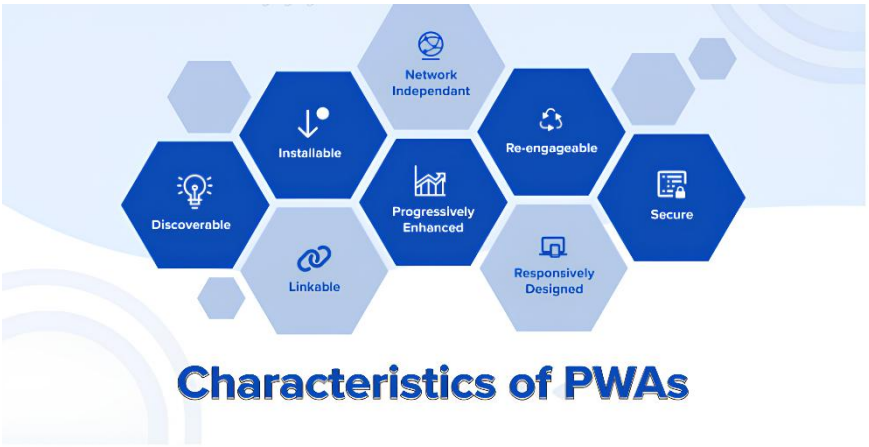
penyimpanan dan sinkronisasi data secara lokal meskipun koneksi internet terbatas [17].

React.js dipilih karena kemampuannya menangani data *real-time* menggunakan arsitektur komponen modular yang fleksibel dan efisien. Teknologi ini memungkinkan visualisasi data dari *drone rover* secara dinamis dalam bentuk grafik, indikator, serta peta interaktif tanpa penurunan performa. Dengan demikian, sistem ini mendukung proses pengambilan keputusan berbasis data terkini di lapangan dengan respons cepat dan akurasi tinggi [18].

2.4 Teknologi *Progressive Web App (PWA)*

Gambar 2.1 mengilustrasikan karakteristik utama *Progressive Web App (PWA)* yang mencerminkan berbagai aspek krusial dalam mendefinisikan keunggulan serta fungsi aplikasi tersebut. Karakteristik ini mencakup kemampuan instalasi pada berbagai perangkat, kemudahan dalam penemuan *(discoverability)*, kemampuan untuk meningkatkan keterlibatan ulang pengguna *(re-engagement)*, tingkat keamanan yang tinggi, serta dukungan terhadap desain responsif.

Seluruh elemen tersebut secara sinergis berkontribusi dalam menciptakan pengalaman pengguna yang lebih unggul, interaktif, dan konsisten lintas platform. Dengan fitur-fitur tersebut, *PWA* tidak hanya menawarkan kinerja yang optimal, tetapi juga memberikan fleksibilitas dalam aksesibilitas, sehingga sangat sesuai untuk diterapkan pada sistem pemantauan pertanian digital yang memerlukan keandalan tinggi meskipun menghadapi keterbatasan konektivitas jaringan.



Gambar 2.1 Karakteristik *Progressive Web App (PWA)* [19]

Gambar 2.1 mempresentasikan konsep *Progressive Web App (PWA)* sebagai suatu pendekatan kontemporer yang mengintegrasikan keunggulan aplikasi *web* dan aplikasi *mobile* ke dalam satu *platform* lintas perangkat. Salah satu keistimewaan utama *PWA* adalah kemampuannya untuk diinstal secara langsung melalui peramban tanpa memerlukan perantara toko aplikasi, sehingga meningkatkan aksesibilitas sekaligus memungkinkan integrasi dengan sistem perangkat, seperti ikon layar utama dan notifikasi [19].

Progressive Web App mendukung operasional secara *offline* melalui implementasi *Service Worker* yang berfungsi menyimpan data secara lokal. Mekanisme tersebut memungkinkan pengguna tetap dapat mengakses konten meskipun tidak tersambung ke jaringan internet, sehingga PWA sangat sesuai digunakan dalam sistem pemantauan drone rover di area perkebunan yang memiliki keterbatasan jaringan. Adanya data lokal, antarmuka yang responsif, dan kemampuan akses berkelanjutan menjadi aspek krusial dalam menjamin keandalan sistem saat digunakan di lapangan.

Gambar 2.1 menampilkan arsitektur beserta interaksi utama antara *Service Worker, cache,* penyimpanan lokal, dan antarmuka pengguna. Struktur ini menjadi fondasi dalam pengembangan sistem pemantauan agrikultur yang efisien, adaptif, dan tahan terhadap gangguan jaringan [20].

Tabel 2.3 Karakteristik *Progressive Web App* (PWA) [20]

No	Karakteristik	Deskripsi Detail
1.	*Discoverable*	*PWA* dirancang agar mudah ditemukan melalui mesin pencari. Setiap aplikasi memiliki *URL* unik yang dapat diindeks oleh mesin pencari,
2.	*Installable*	keunggulan utama *Progressive Web App (PWA)* adalah kemampuannya untuk diinstal langsung pada perangkat pengguna tanpa memerlukan toko aplikasi.
3.	*Linkable*	*Progressive Web App (PWA)* memiliki *URL* unik yang memungkinkan aplikasi untuk dibagikan dengan mudah.
4.	*Progressively Enhanced*	*Web* ini dirancang dengan prinsip desain progresif, yang berarti aplikasi dapat berfungsi dengan baik di berbagai perangkat dan kondisi jaringan.
5.	*Re-engageable*	Fitur *re-engageable*, seperti *push notifications*, memungkinkan aplikasi untuk mengirimkan pemberitahuan kepada pengguna.
6.	*Secure*	Keamanan adalah prioritas utama dalam pengembangan *Progressive Web App (PWA).*
7.	*Network Independent*	*Progressive Web App (PWA)* dapat berfungsi secara offline atau dalam kondisi jaringan yang tidak stabil.
8.	*Responsively Designed*	*Progressive Web App (PWA)* dirancang untuk menyesuaikan tampilan dan fungsionalitasnya sesuai dengan ukuran layar perangkat yang digunakan, baik itu *smartphone*, tablet, atau desktop.

Tabel 2.3 merinci sejumlah karakteristik utama dari *Progressive Web App (PWA)* yang menempatkannya sebagai pilihan teknologi yang sangat relevan dalam pengembangan aplikasi lintas *platform*. Karakteristik ini secara fundamental mendukung kebutuhan akan efisiensi, fleksibilitas, dan keandalan, yang menjadi landasan bagi sistem modern seperti sistem pemantauan berbasis *drone rover*.

PWA bersifat *Discoverable* karena keberadaan *URL* unik yang memungkinkannya diindeks oleh mesin pencari; bersifat *Installable* dengan kemudahan pemasangan langsung dari peramban, melewati toko aplikasi tradisional; serta bersifat *Linkable* yang memudahkan akses dan berbagi melalui tautan.

Sifat *Progressively Enhanced* memungkinkan aplikasi beroperasi pada beragam kondisi jaringan berkat dukungan *Service Worker*; kemampuan *Re-engageable* diwujudkan melalui *push notification* untuk menyampaikan peringatan penting terkait kondisi lahan; dan sifat *Secure* terjamin karena aplikasi berjalan pada protokol *HTTPS* yang melindungi integritas serta keamanan data.

Keunggulan lainnya adalah sifat *Network Independent*, yang memungkinkan aplikasi untuk tetap berfungsi optimal meskipun koneksi internet tidak stabil, dengan mengandalkan sistem *cache* dan penyimpanan lokal. Sementara itu, aspek *Responsive* dari desainnya menjamin tampilan antarmuka dapat menyesuaikan diri dengan sempurna pada berbagai perangkat pengguna.

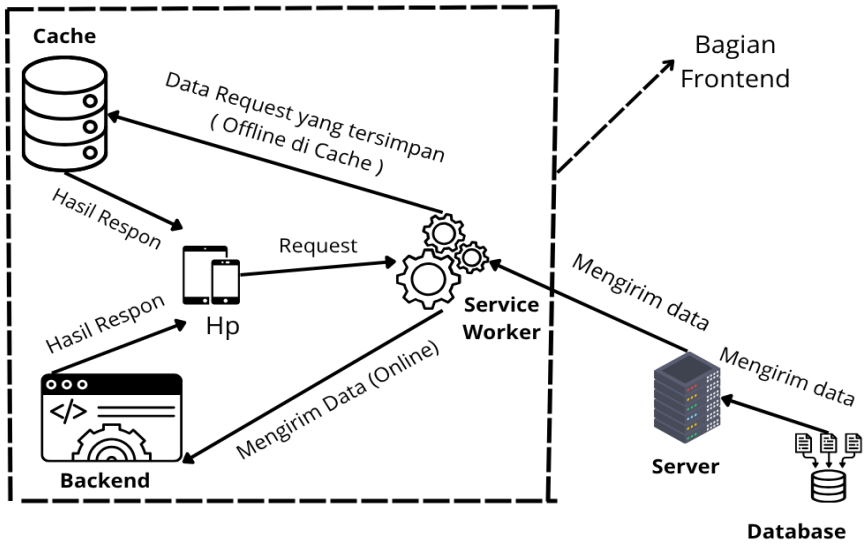
Kombinasi dari seluruh karakteristik ini secara kolektif menjadikan *Progressive Web App (PWA)* sebuah teknologi yang adaptif, aman, dan efisien. Relevansinya menjadi sangat krusial, terutama untuk mendukung implementasi sistem pemantauan pertanian yang berbasis data namun beroperasi di area dengan konektivitas terbatas.

2.4.1 Arsitektur *Progressive Web App (PWA)*

Progressive Web App (PWA) dibangun di atas arsitektur modular yang dirancang untuk menghadirkan pengalaman pengguna yang cepat, adaptif, dan andal, bahkan pada kondisi jaringan terbatas. Arsitektur ini mengintegrasikan komponen utama seperti *Service Worker*, *Web App Manifest*, dan komponen *Frontend* berbasis *JavaScript*. Integrasi tersebut memungkinkan penyimpanan lokal, *caching* konten, serta sinkronisasi data secara efisien [22].

Struktur arsitektur ini memungkinkan aplikasi mempertahankan fungsi utamanya meskipun perangkat tidak terhubung ke internet. Selain itu, rancangan ini memberikan performa tinggi serta kompatibilitas lintas perangkat sehingga mendukung implementasi sistem pemantauan *drone rover* di wilayah perkebunan.

Gambar berikut memperlihatkan visualisasi arsitektur dasar *Progressive Web App (PWA)* beserta hubungan antar komponennya.



Gambar 2.2 Arsitektur *Progressive Web App (PWA)* [22]

Gambar 2.2 memaparkan arsitektur *Progressive Web App (PWA)* yang diimplementasikan pada sistem *Rover Drone*, dengan fokus khusus pada mekanisme *Service Worker*, *Cache*, serta interaksi antara *Frontend* dan *Backend*. Arsitektur ini dirancang untuk menjamin aplikasi tetap operasional, bahkan saat koneksi internet tidak stabil atau terputus sepenuhnya, menjadikannya sangat relevan untuk penerapan di area perkebunan dan wilayah terpencil.

Sisi *frontend* memungkinkan pengguna memperoleh akses ke aplikasi melalui berbagai perangkat seperti ponsel pintar atau tablet tanpa memerlukan instalasi dari toko aplikasi. Pengguna mengajukan permintaan untuk membuka halaman atau mengambil data, permintaan tersebut pertama-tama akan diarahkan pada *Service Worker*. Komponen ini bertindak sebagai perantara krusial antara peramban dan sumber daya *Backend* atau *server*.

Service Worker melakukan pemeriksaan setelah menerima permintaan untuk memastikan apakah data atau aset yang diminta sudah tersimpan di *cache*.

Ketersediaan data memungkinkan Service Worker memberikan respons langsung dari *cache* tanpa harus mengakses jaringan internet. Proses tersebut menjadi mekanisme yang memungkinkan aplikasi tetap berjalan dalam mode offline. Ketidakhadiran data pada *cache* membuat Service Worker meneruskan permintaan tersebut ke server melalui koneksi jaringan (*online request*). Server kemudian melakukan komunikasi dengan basis data untuk memperoleh informasi terbaru.

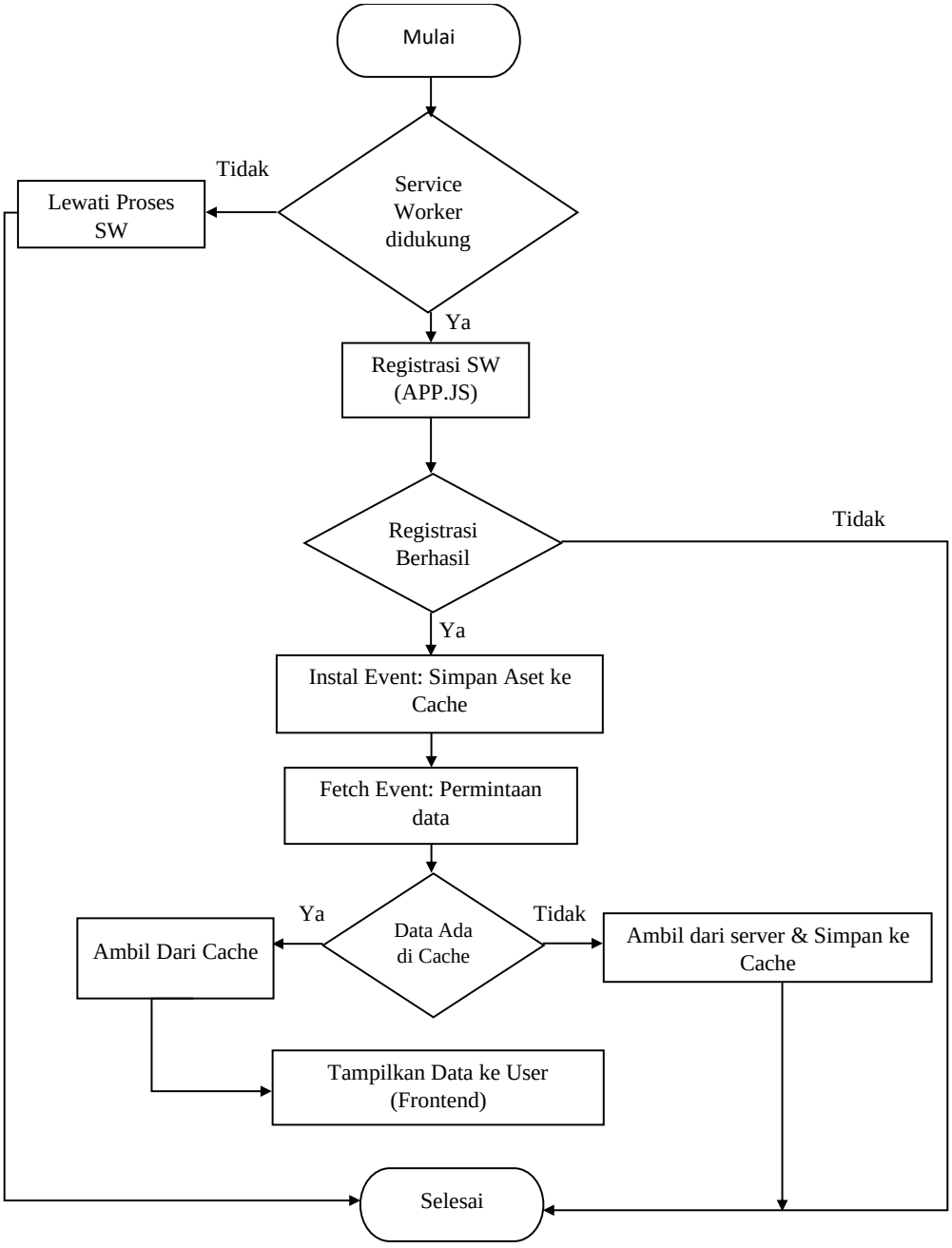
Sisi *Backend* memegang peran penting dalam mengelola logika bisnis aplikasi, termasuk pengolahan data dari server dan penyediaan data untuk antarmuka pengguna *(UI)* yang dinamis. *Backend* berkomunikasi dengan *Service Worker* untuk mengirimkan hasil respons ke perangkat pengguna, baik dalam kondisi *online* maupun *offline*.

Arsitektur ini secara menyeluruh mengadopsi prinsip *offline-first*, di mana aplikasi berupaya memanfaatkan sumber daya lokal (*cache*) terlebih dahulu sebelum bergantung pada jaringan eksternal. Pendekatan ini secara signifikan memperkuat ketahanan sistem terhadap gangguan jaringan dan mengurangi waktu tunggu (*latency*), karena pengguna tidak perlu selalu menunggu respons dari *server*.

Proses pengawasan lahan dan pengendalian rover dapat berjalan secara real-time dan efisien meskipun terdapat fluktuasi konektivitas di lapangan.Lebih dari itu, mekanisme ini juga mengimplementasikan konsep *progressive enhancement*, di mana pengguna dengan koneksi terbatas tetap dapat menikmati sebagian besar fitur utama aplikasi, sementara pengguna dengan koneksi stabil memperoleh pengalaman penuh beserta sinkronisasi *data real-time*.

Implementasi teknis diawali dengan pembuatan berkas *service-worker.js* yang bertanggung jawab dalam menangani proses caching terhadap aset-aset penting aplikasi (seperti HTML, CSS, dan JavaScript utama) serta mengatur prioritas

pengambilan data dari *cache* atau jaringan sesuai dengan kondisi konektivitas



Gambar 2.3 Skema Integrasi *Progressive Web App (PWA)* pada Sistem Monitoring *Drone Rover* [21]

*G*ambar 2.3 mengilustrasikan alur kerja *Service Worker* dalam sebuah aplikasi secara rinci. Proses ini dimulai saat aplikasi dijalankan, kemudian sistem melakukan pengecekan awal untuk memastikan apakah peramban yang digunakan mendukung fitur *Service Worker*. Jika peramban tidak mendukung, sistem akan melewati seluruh proses terkait *Service Worker* dan langsung beralih ke tahap

permintaan data (*fetch event*). Sebaliknya, jika dukungan tersedia, sistem akan melanjutkan ke tahap pendaftaran *Service Worker* yang biasanya diatur dalam berkas *APP.js*.

Proses pendaftaran diikuti oleh pengecekan kedua yang dilakukan sistem untuk memverifikasi keberhasilan registrasi. Kegagalan registrasi membuat aplikasi langsung melanjutkan ke tahap permintaan data tanpa adanya *cache* yang dikelola oleh *Service Worker*. Keberhasilan registrasi akan memicu *install event* pada *Service Worker*. Tahap ini membuat sistem secara proaktif menyimpan aset-aset penting aplikasi, seperti berkas CSS, JavaScript, dan gambar, ke dalam *cache*. Penyimpanan tersebut bertujuan untuk memastikan aplikasi dapat dimuat dengan cepat dan diakses secara offline pada penggunaan berikutnya.

Interaksi pengguna yang memerlukan data dari server akan mengaktifkan *fetch event*. Sistem kemudian melakukan pemeriksaan awal terhadap ketersediaan data yang diminta di dalam *cache*. Ketersediaan data memungkinkan sistem mengambilnya langsung dari *cache* untuk mempercepat waktu muat dan mengurangi beban jaringan. Ketidaktersediaan data membuat sistem mengambilnya dari server sebagai sumber utama. Data yang berhasil diperoleh dari server tidak hanya ditampilkan oleh sistem, tetapi juga disimpan sebagai salinan di dalam *cache* untuk memenuhi permintaan berikutnya.

Data diperoleh baik dari *cache* maupun server maka setelah itu, sistem akan menampilkan informasi tersebut kepada pengguna melalui *frontend*. Keseluruhan alur ini, dari pengecekan dukungan hingga penampilan data, memastikan aplikasi memberikan pengalaman pengguna yang cepat, andal, dan konsisten. Alur kerja cerdas ini menjadi inti dari implementasi *Progressive Web App* yang menjaga sistem tetap fungsional bahkan dalam kondisi jaringan yang tidak stabil.

2.5 *React.Js*

React.js merupakan *library JavaScript* yang dirancang khusus untuk membangun *antarmuka pengguna* (*user interface* atau *UI*) yang interaktif, dinamis, dan adaptif.

Salah satu keunggulan utamanya adalah penerapan pendekatan berbasis komponen, di mana setiap komponen mewakili bagian fungsional dari *UI* yang dapat dipisahkan dan digunakan kembali di berbagai bagian aplikasi [22].

Pendekatan berbasis komponen ini bersifat modular sehingga pengembangan aplikasi menjadi lebih terstruktur dan efisien. Selain itu, sifat *reusable* dari setiap komponen memudahkan pemeliharaan sistem, karena pembaruan pada satu komponen tidak memengaruhi keseluruhan kode program. Dengan demikian, proses pengembangan dapat dilakukan secara bertahap tanpa mengganggu stabilitas sistem.

Tabel 2.4 Arsitektur React.js

No.	Elemen	Deskripsi
1.	Komponen	*React* dibangun di atas konsep komponen yang dapat digunakan kembali, memudahkan pengembangan dan pemeliharaan aplikasi.
2.	*JSX*	*JSX* adalah sintaks yang digunakan untuk mendeskripsikan tampilan *UI*.
3.	*Virtual DOM*	*React* menggunakan Virtual *DOM* untuk meningkatkan performa.
4.	*Lifecycle Methods*	*Lifecycle Methods* memungkinkan pengembang untuk mengelola perilaku komponen pada berbagai tahap.
5.	*Hooks*	*Hooks* adalah fitur baru yang memungkinkan penggunaan state dan efek samping dalam komponen fungsional, menggantikan kebutuhan untuk komponen kelas.

Tabel 2.4 menunjukkan bahwa komponen merupakan elemen dasar dalam arsitektur *React.js*. Setiap komponen menggabungkan logika dan tampilan *UI* secara modular, sehingga memudahkan pengembangan antarmuka yang kompleks.

Tampilan *UI* ditulis menggunakan *JSX (JavaScript XML)*, yaitu sintaks yang memungkinkan elemen *HTML* ditulis langsung dalam *JavaScript*. Pendekatan ini

mempercepat integrasi antara logika program dan visual antarmuka, serta menjadikan kode lebih efisien dan mudah dipelihara.

Fitur penting lain adalah *Virtual DOM*, yang berfungsi mengoptimalkan proses *rendering*. Saat terjadi perubahan data (*state* atau *props*), *React* hanya memperbarui elemen yang berubah tanpa memuat ulang seluruh halaman. Proses ini meningkatkan performa, terutama untuk aplikasi dengan pembaruan data *real-time*.

Dalam manajemen data, *state* berfungsi sebagai data internal yang berubah secara dinamis, sedangkan *props* digunakan untuk meneruskan data antarkomponen. Aliran data satu arah (*unidirectional data flow*) menciptakan sistem *UI* yang stabil dan mudah diprediksi.

Lifecycle methods memberikan fleksibilitas bagi pengembang untuk mengeksekusi logika tertentu pada fase inisialisasi, pembaruan, atau penghapusan komponen. Kombinasi struktur komponen, JSX, Virtual DOM, serta pengelolaan *state* dan *props* menjadikan React.js sebagai arsitektur yang efisien dan ideal untuk aplikasi web modern.

Tabel 2.5 Keuntungan Menggunakan *React.js* [23]

No.	Keuntungan	Deskripsi
1.	*Reusable Components*	Komponen yang dapat digunakan kembali mengurangi duplikasi kode dan meningkatkan efisiensi pengembangan.
2.	*Performance*	Penggunaan Virtual *DOM* meningkatkan kinerja aplikasi dengan meminimalkan interaksi langsung dengan *DOM*.
3.	*Declarative UI*	Pendekatan deklaratif membuat kode lebih mudah dibaca dan dipahami, serta memudahkan *debugging*.
4.	*Strong Community*	*React* memiliki komunitas yang besar dan aktif, menyediakan banyak sumber daya, pustaka, dan dukungan.

5.	*Flexibility*	*React* dapat digunakan dengan berbagai pustaka dan *framework* lain, memberikan fleksibilitas dalam pengembangan.

Tabel 2.5 menyoroti keunggulan utama *React.js* dalam pengembangan antarmuka pengguna *(UI).* Salah satu keunggulan utamanya adalah konsep *reusable components*, yaitu kemampuan membangun elemen *UI* yang modular dan dapat digunakan kembali di berbagai bagian aplikasi. Pendekatan ini mempercepat pengembangan sekaligus menjaga konsistensi tampilan dan perilaku sistem.

Dari sisi performa, React memanfaatkan Virtual *DOM* untuk mengoptimalkan proses *rendering*. Dengan membandingkan perubahan antara versi Virtual *DOM* lama dan baru, *React* hanya memperbarui elemen yang berubah pada *DOM* aktual. Mekanisme ini meningkatkan efisiensi dan mengurangi beban kerja browser.

React menerapkan pendekatan declarative UI, di mana tampilan ditentukan oleh *state* aplikasi. Pendekatan ini membuat kode lebih sederhana, mudah dibaca, dan mengurangi potensi kesalahan logika tampilan.

React juga didukung oleh komunitas global yang luas dan aktif, menyediakan dokumentasi, pustaka tambahan, serta pembaruan rutin. Dukungan ini memperkuat ekosistem *React.js* sebagai kerangka kerja *front-end* modern yang efisien, fleksibel, dan berorientasi pada kinerja tinggi.

Tabel 2.6 Kerugian Menggunakan *React.js* [23]

No.	Kerugian	Deskripsi
1.	*Learning Curve*	Meskipun konsep dasar mudah dipahami, fitur lanjutan seperti *hooks* dan *state management* dapat memiliki *kurva* belajar yang curam.
2.	*Boilerplate Code*	Pengaturan awal dan konfigurasi dapat memerlukan banyak kode *boilerplate*, untuk aplikasi besar.
3.	*SEO Challenges*	Meskipun ada solusi untuk *SEO*, aplikasi *React* yang sepenuhnya berbasis *client-side* dapat menghadapi tantangan dalam pengindeksan oleh mesin pencari.

4.	*Frequent Updates*	*React* sering diperbarui, yang dapat menyebabkan ketidak cocokan dengan pustaka lain.

Tabel 2.6 memaparkan berbagai kendala yang melekat pada *React.js*. Tingkat kesulitan awal yang terbilang curam menjadi salah satu tantangan paling menonjol, suatu kondisi yang sangat terasa bagi para pengembang pemula. Meski didasari oleh komponen modular, namun pemahaman yang mendalam mengenai konsep seperti *state management*, *props*, dan arsitektur komponen itu sendiri justru memerlukan waktu serta proses pembelajaran yang intensif.

React juga cenderung mengharuskan adanya *boilerplate code* dalam volume yang tidak kecil pada tahapan awal dari inisialisasi sebuah proyek. Kegiatan-kegiatan seperti integrasi pustaka pendukung dan konfigurasi lingkungan *build* ini seringkali justru menambah tingkat kompleksitas, sebuah permasalahan yang kentara terutama pada proyek-proyek bersekala besar.

Sudut pandang *Search Engine Optimization (SEO), React* dihadapkan pada sebuah kendala bersifat struktural, di mana konten yang dirender secara dinamis melalui *client-side rendering* acap kali menyulitkan proses pengindeksan oleh mesin pencari. Untuk mengatasi kendala ini, pendekatan *Server-Side Rendering (SSR)* atau *pre-rendering* pun menjadi solusi yang umumnya diterapkan.

2.6 *Media query*

Media query merupakan teknik penting dalam pengembangan *antarmuka pengguna* yang adaptif terhadap berbagai ukuran layar dan jenis perangkat. Dalam konteks *Progressive Web App (PWA)* berbasis *React.js, media query* memungkinkan perancangan tata letak yang responsif tanpa ketergantungan pada kerangka kerja tambahan seperti *Bootstrap*.

Menurut Kim et al. (2020) dalam penelitian berjudul *"Multi-Device Complementary View Adaptation with Liquid Media Queries", media query* tidak hanya menyesuaikan dimensi layar, tetapi juga mempertimbangkan orientasi, resolusi, dan kemampuan visual perangkat. Dengan demikian, antarmuka dapat beradaptasi secara *real-time* terhadap berbagai kondisi perangkat [24].

Konsep *Liquid Media Queries* memperluas batasan responsivitas tradisional dengan menyesuaikan tampilan di beberapa perangkat secara simultan. Pendekatan ini memungkinkan sinkronisasi antartampilan, meningkatkan kenyamanan pengguna, serta mempertahankan konsistensi visual di berbagai perangkat.

Integrasi *media query* dalam *React.js* dapat dilakukan melalui berbagai metode, seperti *styled-components*, *CSS Modules*, atau pustaka serupa yang mendukung gaya per komponen. Pendekatan tersebut sejalan dengan arsitektur modular *React*, yang mengutamakan efisiensi dan keteraturan struktur kode.

M*edia query* tetap berfungsi dalam mode *offline* berkat dukungan *service worker*. Dengan dukungan ini, tampilan aplikasi tetap responsif meskipun koneksi internet tidak tersedia. Pendekatan ini sangat penting dalam pengembangan *Progressive Web App (PWA)* untuk pemantauan di area dengan jaringan terbatas.

Penerapan *media query* memberikan fondasi adaptif yang kuat dan efisien dalam membangun *user interface* progresif yang responsif, fleksibel, serta berorientasi pada pengalaman pengguna dalam ekosistem *React.js*.

Tabel 2.7 Keuntungan *Media Query* dalam *Front-End PWA* berbasis *React.js* [24]

No	Aspek	Keuntungan	Penjelasan Singkat Berdasarkan Jurnal
1	Responsivitas	Mendukung tampilan lintas perangkat	*Media query* memungkinkan penyesuaian *UI* terhadap berbagai resolusi dan orientasi layar (mobile, tablet, desktop).
2	Adaptasi Konteks *Multi-Device*	Menyediakan tampilan pelengkap antar perangkat	*Liquid Media Query* memungkinkan tampilan berbeda namun saling melengkapi antar dua atau lebih perangkat secara bersamaan.
3	Efisiensi Kode	Mengurangi ketergantungan pada *framework UI* eksternal	Penggunaan media *query* langsung di *React.js* mengurangi

			kebutuhan akan *Bootstrap* dan *framework* lain.
4	Pengalaman Pengguna *(UX)*	Meningkatkan kenyamanan dan konsistensi tampilan	Tampilan *UI* dapat dirancang lebih fokus dan adaptif sesuai *device* yang digunakan pengguna.
5	Skalabilitas	Mudah diperluas dan diintegrasikan dalam komponen *React*	*Media query* dapat diintegrasikan melalui *styled-components*, emotion, atau *CSS* module pada *React* untuk skalabilitas.
No	Aspek	Keuntungan	Penjelasan Singkat Berdasarkan Jurnal
6	Dukungan *Offline & Cache*	Tetap responsif meskipun dalam mode offline *Progressive Web App (PWA)*	Responsivitas tidak terganggu walaupun aplikasi dijalankan secara offline berkat *service worker + media query.*
7	Responsivitas	Mendukung tampilan lintas perangkat	*Media query* memungkinkan penyesuaian *UI* terhadap berbagai resolusi dan orientasi layar (mobile, tablet, desktop) secara otomatis.
8	Adaptasi Konteks *Multi-Device*	Menyediakan tampilan pelengkap antar perangkat	*Liquid Media Query* memungkinkan tampilan berbeda namun saling melengkapi antar dua atau lebih perangkat.

Tabel 2.7 memaparkan berbagai keuntungan dari penerapan media query dalam pengembangan *front-end* Progressive Web App (PWA) yang menggunakan React.js, sebuah sintesis yang merujuk pada penjelasan dari Kim et al. (2021) dalam jurnal mereka *"Multi-Device Complementary View Adaptation with Liquid Media Queries".*

Keunggulan utama dari pendekatan ini terletak pada kemampuannya untuk menciptakan antarmuka pengguna *(UI)* yang mampu beradaptasi terhadap berbagai perangkat, yang memungkinkan tampilan untuk menyesuaikan diri secara otomatis

terhadap variasi ukuran dan orientasi layar. Kemampuan adaptif ini secara otomatis menghilangkan kebutuhan untuk membangun versi aplikasi yang terpisah untuk setiap *platform,* yang pada akhirnya menyederhanakan keseluruhan proses pengembangan [24].

Integrasi media *query* ke dalam komponen *React* dapat diwujudkan melalui berbagai metode, seperti *styled-components, CSS Modules,* atau pustaka-pustaka serupa lainnya yang mendukung pemisahan gaya per komponen. Pendekatan ini sejalan erat dengan filosofi arsitektur modular yang dianut *React*, yang mana sangat menekankan pada efisiensi, keteraturan struktur kode, serta kemudahan dalam pemeliharaan antarmuka.

Lebih jauh lagi, keunggulan signifikan dari *media query* juga bersumber dari kompatibilitasnya dengan sistem yang beroperasi dalam mode *offline*. Dukungan ini menjamin kontinuitas pengalaman pengguna, bahkan pada saat koneksi internet sedang tidak tersedia. Aspek ini menjadi sangat krusial, terutama dalam konteks pengembangan *PWA* yang ditujukan untuk wilayah-wilayah dengan cakupan jaringan terbatas.

Tabel 2.8 Karakteristik *Media Query* dalam *Front-End PWA + React.js I* [24]

No	Karakteristik	Deskripsi Singkat
1	*Responsiveness*	Menyesuaikan tampilan berdasarkan lebar, tinggi, orientasi, dan resolusi layar.
2	Adaptivitas Kontekstual	Memungkinkan antarmuka menyesuaikan dengan kondisi perangkat pengguna secara dinamis.
3	Modularitas	Dapat diintegrasikan pada level komponen *(component-level styling*) di *React.*
4	Skalabilitas	Memudahkan pengembangan berkelanjutan dengan penyesuaian antar *device.*
5	Interoperabilitas	Kompatibel dengan berbagai browser modern dan mendukung teknologi *Progressive Web App (PWA).*
6	Performa Ringan	Mengurangi ketergantungan pada *framework* eksternal seperti *Bootstrap.*

7	*Maintainability*	Gaya *(style)* dapat dikelola secara terpisah dalam file *CSS* atau *styled-components*.
8	Reusabilitas	Dapat digunakan ulang di berbagai komponen tanpa menyalin ulang banyak aturan *CSS*.
9	Integrasi dengan *JS Framework*	Mudah dihubungkan dengan *React* (misalnya, melalui *styled-components*, *Emotion*, dll).
10	Mendukung *Offline-First*	Tetap bekerja meskipun aplikasi dalam mode *offline*

Tabel 2.8 menjelaskan karakteristik utama *media query* dalam pengembangan *user interface* untuk *Progressive Web App (PWA)* berbasis *React.js*. Salah satu karakteristik penting adalah konsep *responsiveness*, yakni kemampuan antarmuka untuk secara otomatis menyesuaikan tata letak dan skala tampilan terhadap berbagai ukuran layar tanpa perlu memuat ulang halaman. Kemampuan ini memastikan tampilan antarmuka tetap optimal pada perangkat *mobile*, *tablet*, maupun *desktop*.

M*edia query* mendukung konsep *adaptivitas kontekstual* yang memungkinkan sistem menyesuaikan tampilan dengan kondisi perangkat, seperti orientasi layar, resolusi, serta kemampuan visual. Dengan demikian, pengguna memperoleh pengalaman interaksi yang konsisten dan intuitif di berbagai perangkat.

Karakteristik lain seperti interoperabilitas, efisiensi performa, dan kompatibilitas lintas *peramban* semakin menegaskan peran penting *media query* dalam pengembangan antarmuka berbasis komponen. Integrasinya dengan pendekatan *offline-first* pada *PWA* juga meningkatkan keandalan sistem, karena tampilan tetap dapat diakses meskipun koneksi internet terbatas.
Dari sisi pengelolaan kode, *media query* mendukung prinsip *maintainability* dan *reusability* melalui pemisahan logika desain dengan *file CSS* eksternal atau pendekatan *CSS-in-JS* seperti *styled-components*. Strategi ini mendorong pengembangan antarmuka yang modular, terstruktur, dan konsisten di seluruh komponen sistem.

Dengan demikian, *media query* memberikan fondasi adaptif yang kuat dan efisien dalam membangun antarmuka progresif yang responsif, fleksibel, serta berorientasi pada pengalaman pengguna di dalam ekosistem pengembangan *React.js*.

2.7 Pengelolaan data yang diolah pada tampilan *web*

Dalam sistem *drone rover* untuk pemantauan dan pemeliharaan perkebunan kelapa sawit, antarmuka pengguna *(frontend)* menerima data dari *backend* dalam format *JavaScript Object Notation (JSON)*. Format *JSON* merupakan bentuk pertukaran data berbasis teks yang ringan serta mudah dibaca, baik oleh manusia maupun oleh mesin. Struktur *JSON* terdiri atas pasangan *key value*, di mana setiap *key* mewakili atribut tertentu, sedangkan *value* dapat berupa teks, angka, *array*, maupun objek terstruktur [25].

Penggunaan *JSON* mendukung pengolahan data secara efisien pada sistem berbasis *Progressive Web App (PWA)*. Format ini memudahkan proses penyajian informasi melalui *dashboard*, pengelolaan interaksi pengguna, serta integrasi dengan layanan eksternal, seperti sistem pemetaan berbasis lokasi dan analisis data sensor.

Selain itu, *JSON* juga berperan penting dalam menjembatani komunikasi antara antarmuka pengguna dan sistem pengelolaan data *real-time*. Dengan struktur yang sederhana namun fleksibel, format ini memungkinkan pertukaran data berlangsung cepat dan konsisten.

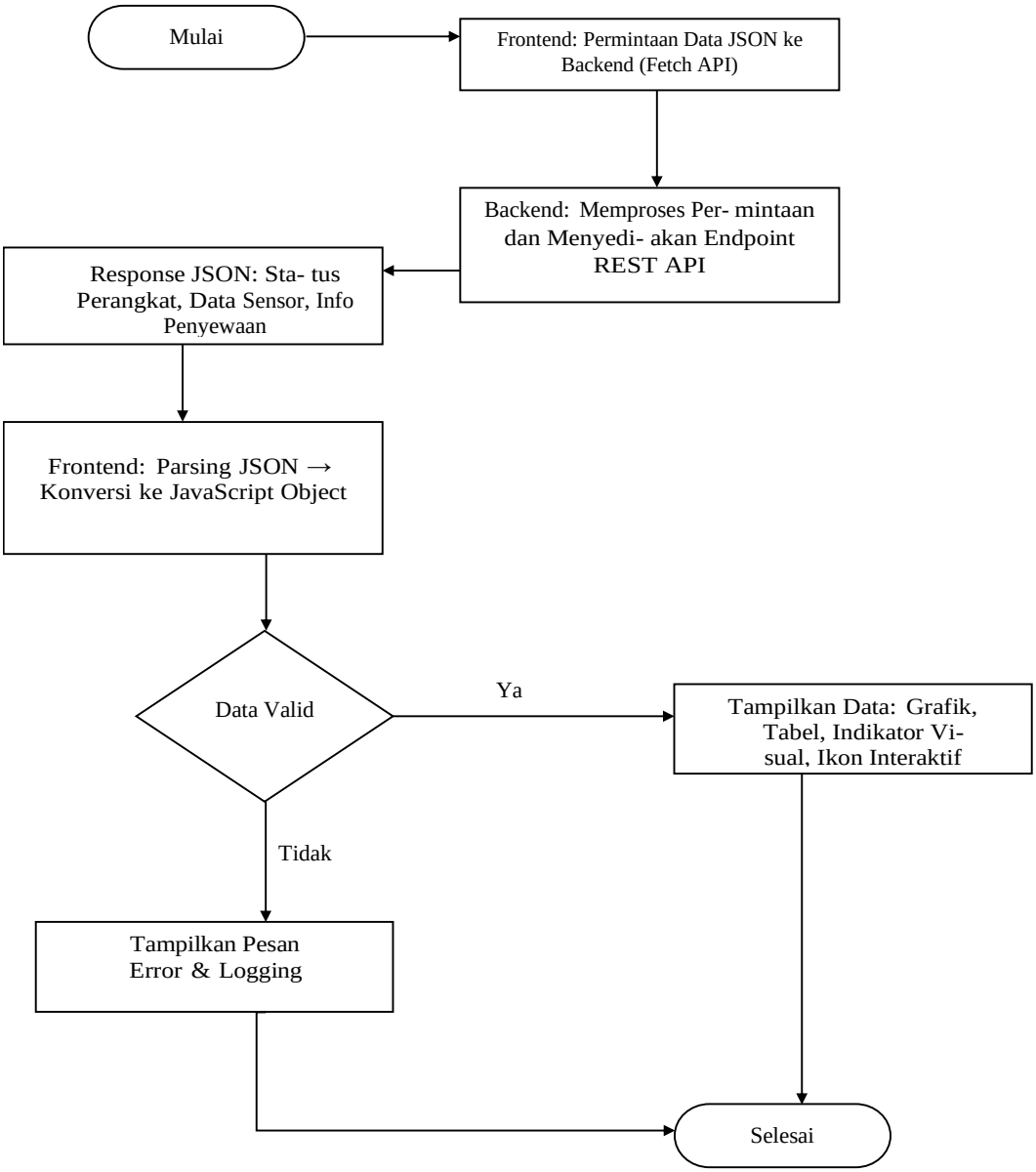
Dengan demikian, *JSON* menjadi format data yang andal dalam mendukung interoperabilitas dan efisiensi sistem monitoring agrikultur digital berbasis *drone rover* dan *Progressive Web App (PWA)*.

2.7.1 Struktur dan Alur Pengambilan Data *JSON*

Gambar 2.4 memaparkan diagram alur yang merinci proses pengambilan dan pengolahan data *JSON* pada sistem pemantauan drone rover berbasis *Progressive Web App (PWA)*. Diagram ini secara visual menggambarkan seluruh tahapan,

dimulai dari saat permintaan data dibuat oleh *frontend* hingga informasi akhir disajikan kepada pengguna.

Visualisasi ini berfungsi sebagai panduan komprehensif untuk memahami integrasi yang terjadi antara backend dan *frontend*. Selain itu, ia juga menjelaskan hubungan logis yang mengikat setiap proses di dalam sistem secara keseluruhan.
Dengan demikian, alur dari pengambilan, validasi, hingga penyajian data dapat dipahami secara lebih terstruktur dan sistematis. Akibatnya, proses komunikasi antar lapisan sistem menjadi berjalan efisien dan jauh lebih mudah untuk dianalisis.



Gambar 2.4 Diagram Alur Proses Permintaan, Validasi, dan Penyajian Data *JSON* pada Sistem *PWA* [25]

Gambar 2.4 menunjukkan alur proses pengambilan dan penyajian data *JSON* pada sistem *Progressive Web App (PWA)* untuk pemantauan *drone rover*. Proses dimulai ketika *frontend* mengirimkan permintaan data dalam format *JSON* ke *backend* menggunakan *Fetch API.*

Backend memproses permintaan tersebut dan menyediakan *endpoint* REST API yang berisi data status perangkat, data sensor, serta informasi penyewaan. Proses pemrosesan yang telah selesai membuat backend mengirimkan respons dalam format JSON kembali ke *frontend* [26].

Sistem pada sisi *frontend* melakukan *parsing* terhadap data JSON untuk mengonversinya menjadi objek JavaScript. Sistem kemudian melaksanakan proses validasi data guna memastikan kelengkapan dan ketepatan informasi yang diterima [27].

Sistem akan menyajikan informasi tersebut melalui grafik, tabel, indikator visual, atau ikon interaktif apabila data dinyatakan valid sehingga pengguna dapat memahami data secara *real time*. Sistem menampilkan pesan kesalahan serta melakukan *logging* apabila data tidak valid untuk mendukung proses *debugging* dan perbaikan pada tahap berikutnya.

Sistem memanfaatkan mekanisme ini untuk memastikan bahwa integrasi antara frontend dan backend berlangsung secara terstruktur serta efisien. Konsekuensinya, proses penyajian data secara *real time* menjadi lebih akurat, konsisten, dan mudah dipahami oleh pengguna.

2.7.2 Pentingnya Format *JSON* dalam *Frontend*

Penggunaan format *JSON* sebagai standar pertukaran data antara *backend* berbasis *Express.js* dan *frontend React.js* memberikan fleksibilitas tinggi dalam integrasi antar lapisan sistem. Format ini memiliki struktur ringan berbasis pasangan *key–*

value dan mudah dikonversi menjadi *state* dalam *React*, sehingga mendukung pengembangan antarmuka pengguna yang dinamis, modular, dan responsif [28].

JSON unggul dalam hal interoperabilitas karena dapat dipahami oleh hampir semua bahasa pemrograman modern. Hal ini menjadikannya pilihan ideal untuk komunikasi data lintas platform. Dalam konteks aplikasi *Progressive Web App* (*PWA*) berbasis *React.js*, data yang diterima dari *backend* dalam format *JSON* dapat langsung diintegrasikan dengan komponen *React* untuk menghasilkan tampilan visual yang interaktif, seperti grafik, tabel, dan indikator status perangkat secara *real time*.

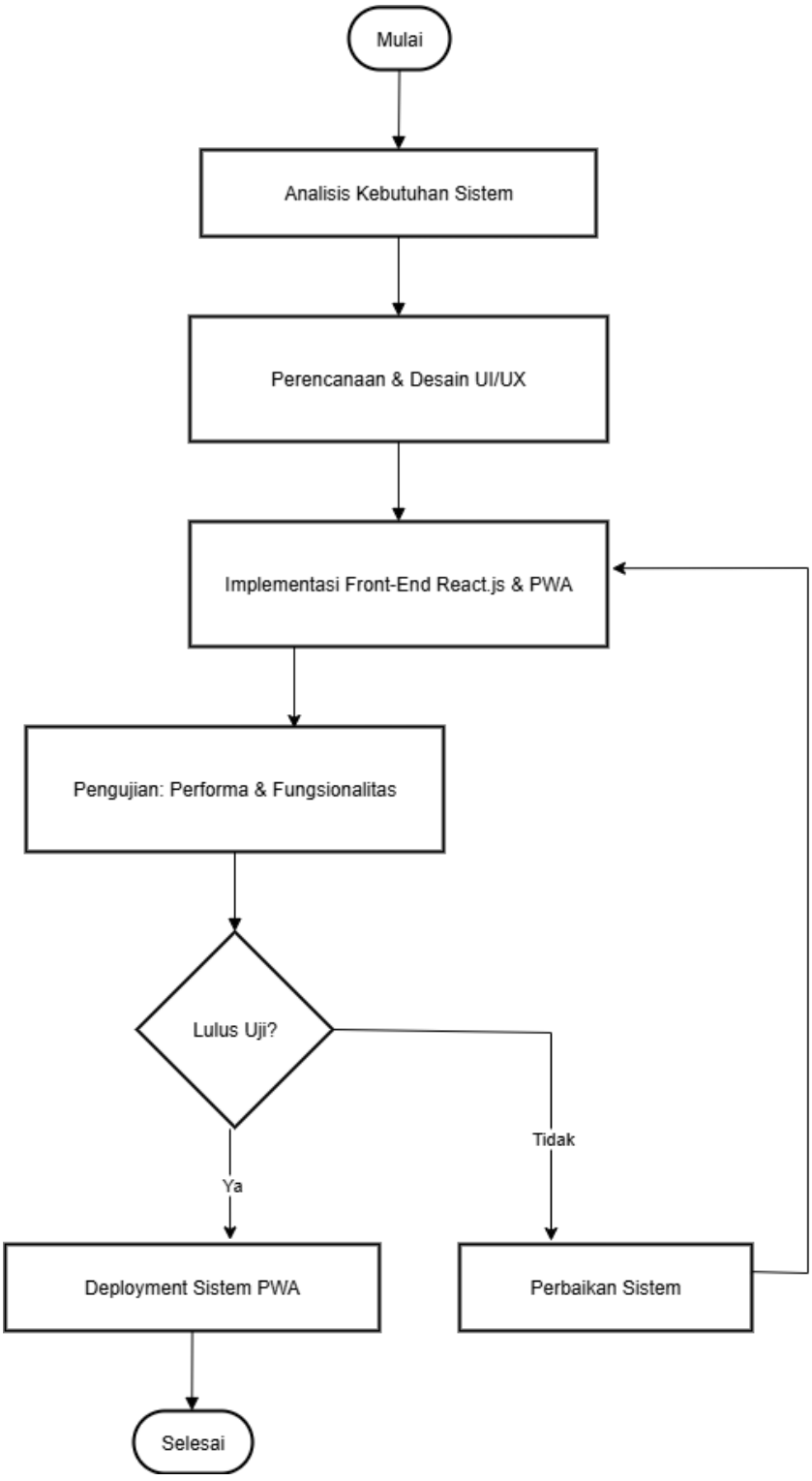
Pemanfaatan *JSON* secara optimal memungkinkan aplikasi *PWA* menyajikan pengalaman pengguna yang interaktif sekaligus efisien, khususnya dalam pemantauan dan pemeliharaan perkebunan kelapa sawit menggunakan sistem *drone rover*. Format ini juga memfasilitasi pertukaran data yang hemat *bandwidth* namun tetap terstruktur dan mudah dikelola oleh sistem antarmuka.

Kondisi tersebut sangat relevan pada wilayah dengan keterbatasan konektivitas jaringan, karena data yang telah diambil dapat disimpan sementara di sisi *frontend* dan diperbarui secara otomatis ketika koneksi internet kembali tersedia. Dengan demikian, penggunaan *JSON* tidak hanya berperan sebagai jembatan komunikasi antara *backend* dan *frontend*, tetapi juga sebagai elemen penting dalam mendukung rancangan aplikasi yang adaptif, efisien, serta andal untuk pemantauan agrikultur berbasis data *real-time*.

2.8 Penggunaan Metode *Kanban*

Konsep utama dari metode Kanban adalah visualisasi alur kerja. Teknik ini digunakan untuk menggambarkan proses kerja secara grafis sehingga tim dapat memantau status setiap tugas secara *real-time*. Istilah *real-time* merujuk pada kemampuan sistem dalam menampilkan informasi terkini tanpa penundaan [29].

Berikut merupakan diagram alir penggunaan metode *Kanban*:



Gambar 2.5 Penggunaan Metode Kanban [29]

Diagram alir pada Gambar 2.5 menggambarkan tahapan proses penelitian yang dimulai dari titik inisiasi (Mulai) sebagai langkah awal pelaksanaan kegiatan. Tahap pertama yang dilakukan adalah Analisis Kebutuhan Sistem, yang bertujuan mengidentifikasi kebutuhan fungsional dan nonfungsional secara menyeluruh. Proses ini mempertimbangkan aspek operasional serta konteks teknis yang relevan, sehingga dapat menghasilkan spesifikasi sistem yang jelas dan terukur.

Tim pengembang melanjutkan proses ke tahap Perencanaan dan Desain UI/UX setelah kebutuhan sistem terdefinisi dengan baik. Tahap ini menyusun rancangan antarmuka pengguna dan pengalaman pengguna secara sistematis serta terstruktur. Desain yang dihasilkan menjadi dasar utama dalam pengembangan sistem agar tetap selaras dengan kebutuhan pengguna dan tujuan penelitian.

Tahap berikutnya adalah Implementasi *Frontend React.js & PWA*, yang merepresentasikan proses pengkodean serta pengembangan sistem berdasarkan rancangan yang telah dibuat sebelumnya. Implementasi ini mengacu pada standar pengembangan perangkat lunak agar setiap komponen antarmuka dan fungsionalitas utama dapat berjalan secara optimal.

Sistem memasuki tahap Pengujian: Performa dan Fungsionalitas setelah proses implementasi selesai untuk memastikan bahwa seluruh komponen berfungsi sesuai spesifikasi. Pengujian ini bertujuan menilai performa, keandalan, serta stabilitas sistem dalam kondisi operasional yang menyerupai penggunaan nyata.

Apabila hasil pengujian menunjukkan keberhasilan, sistem akan melalui tahap Lulus Uji. Dari sini, proses dilanjutkan ke *Deployment* Sistem *PWA*, yaitu penerapan sistem ke lingkungan produksi agar dapat digunakan secara aktual. Sebaliknya, apabila sistem tidak lulus pengujian, dilakukan tahap *Perbaikan Sistem* guna menyempurnakan komponen yang belum memenuhi kriteria sebelum diuji kembali.

Seluruh rangkaian proses berakhir pada tahap Selesai, yang menandai tuntasnya kegiatan penelitian sekaligus menjadi dasar penyusunan kesimpulan dan saran dalam laporan akhir. Dengan demikian, metode *Kanban* memberikan alur kerja

yang terstruktur, transparan, dan efisien dalam mendukung keberhasilan penelitian serta pengembangan sistem berbasis *PWA*.

2.9 Visual Studio Code

Visual Studio Code (*VS Code*) versi 1.80.1 merupakan penyunting kode sumber lintas platform yang dikembangkan oleh *Microsoft*. Aplikasi ini dapat dijalankan pada sistem operasi *Windows*, *Linux*, dan *macOS*. Editor ini mendukung berbagai bahasa pemrograman seperti *HTML*, *CSS*, *JavaScript*, *PHP*, dan *Python* [30]. Beberapa fitur utamanya mencakup *debugging*, integrasi *Git*, penyorotan sintaksis, *auto-complete*, *snippet*, serta *code refactoring* yang membantu pengembang menulis dan mengelola kode secara efisien.

Open-source VS Code menawarkan fleksibilitas tinggi bagi pengembang namun, distribusi resminya tetap menggunakan lisensi *proprietary* untuk menjaga konsistensi kualitas dan keberlanjutan pengembangan. Editor ini dibangun menggunakan *Electron*, yaitu kerangka kerja berbasis *Node.js* dan mesin *Blink* yang memungkinkan pembuatan aplikasi lintas platform dengan antarmuka modern.

Integrasi teknologi tersebut memberikan kemampuan penyuntingan kode yang lebih canggih, ringan, dan responsif. Dengan demikian, *Visual Studio Code* mendukung produktivitas pengembang dalam membangun aplikasi berbasis web maupun sistem modern lainnya secara efisien dan profesional.

III. METODE PENELITIAN

3.1. Waktu dan Tempat

Penelitian ini dilaksanakan di Laboratorium Teknik Elektro, Jurusan Teknik Elektro, Universitas Lampung, selama enam bulan dengan tahapan yang terstruktur. Kegiatan diawali dengan studi literatur pada bulan pertama hingga bulan kelima sebagai dasar teoritis. Selanjutnya, penyusunan rencana dan pelaksanaan seminar proposal dilakukan pada bulan kedua.

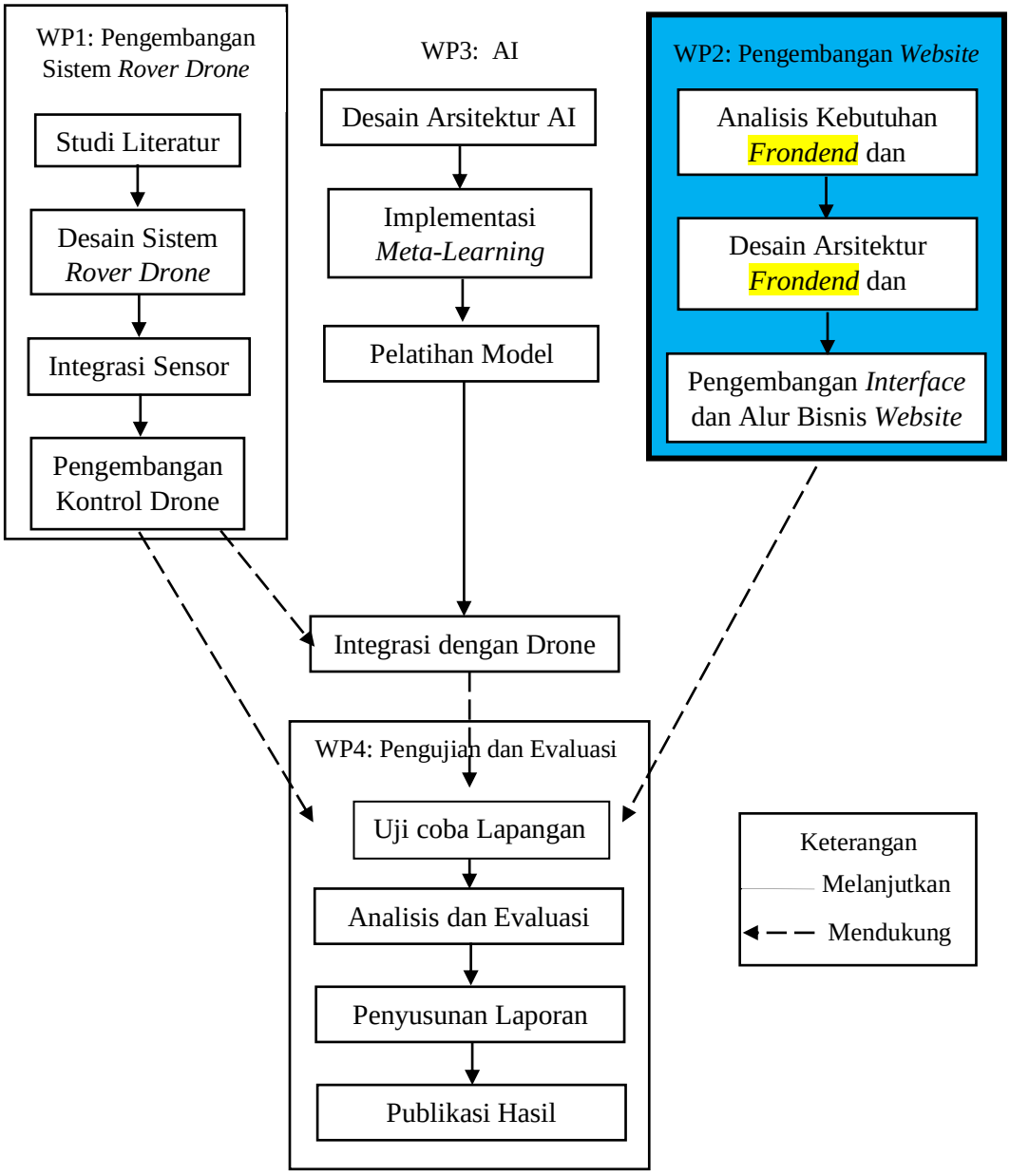
Tahap pengembangan sistem (*Work in Progress*) berlangsung sejak akhir bulan kedua hingga bulan kelima, dengan fokus pada implementasi *React.js* dan fitur *Progressive Web App (PWA)* seperti *Service Worker* dan konsep *offline-first*. Penulisan laporan dilakukan bersamaan dengan proses pengujian sistem pada bulan kelima, kemudian dilanjutkan dengan tahap penyelesaian (*Done*) dan seminar hasil pada akhir bulan keenam.

Perencanaan waktu yang sistematis maka penelitian ini diharapkan dapat diselesaikan tepat waktu serta memberikan kontribusi nyata terhadap peningkatan efisiensi sistem pemantauan perkebunan kelapa sawit berbasis teknologi *web* progresif.

3.2 Cakupan Penelitian

Penelitian ini merupakan bagian dari proyek besar bernama *Rover Drone*, yang dirancang untuk meningkatkan produktivitas perkebunan kelapa sawit melalui pemanfaatan teknologi *drone* dan *rover*. Dalam proyek ini, diterapkan sistem

terintegrasi yang meliputi pengumpulan data sensor lingkungan, pengendalian perangkat otomatis, serta penyimpanan dan pengelolaan data berbasis *web*. Seluruh sistem tersebut bertujuan untuk meningkatkan efisiensi operasional.



Gambar 3.1 Cakupan Penelitian

Gambar 3.1 merupakan cakupan penelitian berfokus pada *Work Package (WP) 2:* Pengembangan *Website* sebagai bagian utama dari keseluruhan proyek. WP2 mencakup tahapan analisis kebutuhan komponen *frontend* dan *backend*, perancangan arsitektur sistem, serta implementasi antarmuka pengguna dan alur

bisnis aplikasi. Penelitian ini menitikberatkan pada pengembangan *frontend* yang terintegrasi dengan sistem *Rover Drone* untuk mendukung proses pemantauan berbasis *web* secara efisien.

3.3 Alat dan Bahan

Alat dan bahan yang digunakan pada penelitian kali ini adalah sebagai berikut:

3.3.1 Alat

Alat-alat yang digunakan dalam penelitian ini meliputi:

1. Koneksi Internet (Wi-Fi atau jaringan seluler)

   Fungsi: Diperlukan untuk sinkronisasi data *real-time* antara perangkat di lapangan (*drone*, *rover*, dan sensor) dengan *server* pusat.

2. Laptop

   Fungsi: Digunakan sebagai perangkat utama untuk menjalankan perangkat lunak pengembangan, menulis kode, serta menguji aplikasi yang dikembangkan.

3. Peramban Web (*Chrome* atau *Firefox*)

   Fungsi: Berperan dalam menguji tampilan dan fungsionalitas sistem. Selain itu, peramban digunakan untuk melakukan inspeksi elemen dan analisis performa.

4. Penyunting Kode (*Visual Studio Code*)

   Fungsi: Digunakan untuk menulis kode *React.js*, mengelola struktur proyek, serta menjalankan proses *debugging*. *Visual Studio Code* dipilih karena memiliki fitur lengkap dan mendukung ekstensi pengembangan web.

5. Perangkat Dokumentasi (*Microsoft Word* atau *Google Docs*)

   Fungsi: Dimanfaatkan untuk penulisan laporan skripsi dan dokumentasi teknis selama pengembangan sistem.

3.3.2 Bahan

Bahan-bahan yang digunakan bersifat perangkat lunak dan sumber daya digital, antara lain:

1. Kerangka kerja *React.js* (melalui *Yarn*)

   Fungsi: Menjadi pustaka *JavaScript* utama untuk membangun antarmuka pengguna secara modular dan responsif. Instalasi serta manajemen dependensi dilakukan menggunakan *package manager Yarn*.

2. Pustaka Pendukung (*Library*)

   Fungsi: Digunakan untuk mendukung navigasi halaman, komunikasi *API*, dan animasi antarmuka guna meningkatkan pengalaman pengguna.

3. Konfigurasi *Progressive Web App (PWA)*

   Fungsi: Dilakukan melalui penambahan berkas *manifest.json* dan *service-worker.js* agar aplikasi dapat berjalan secara *offline* serta diinstal layaknya aplikasi *native*.

4. Data atau *API* dari *Backend*

   Fungsi: Digunakan untuk menampilkan data sensor seperti suhu, kelembapan, dan intensitas cahaya secara *real-time* pada antarmuka. Pada tahap penelitian ini, sebagian besar data masih berupa *dummy/*simulasi, dengan satu set data nyata sebagai validasi awal. Pendekatan ini memungkinkan sistem diuji dan dievaluasi meskipun data lapangan belum sepenuhnya tersedia.

3.4. Analisa Kebutuhan

Tahap analisis kebutuhan merupakan fase awal yang krusial dalam pengembangan sistem pemantauan perkebunan kelapa sawit berbasis teknologi digital. Proses ini mencakup identifikasi kebutuhan fungsional dan teknis, seperti integrasi perangkat keras dan perangkat lunak, perancangan antarmuka pengguna, serta penyediaan infrastruktur pendukung agar sistem dapat beroperasi secara optimal di lapangan.

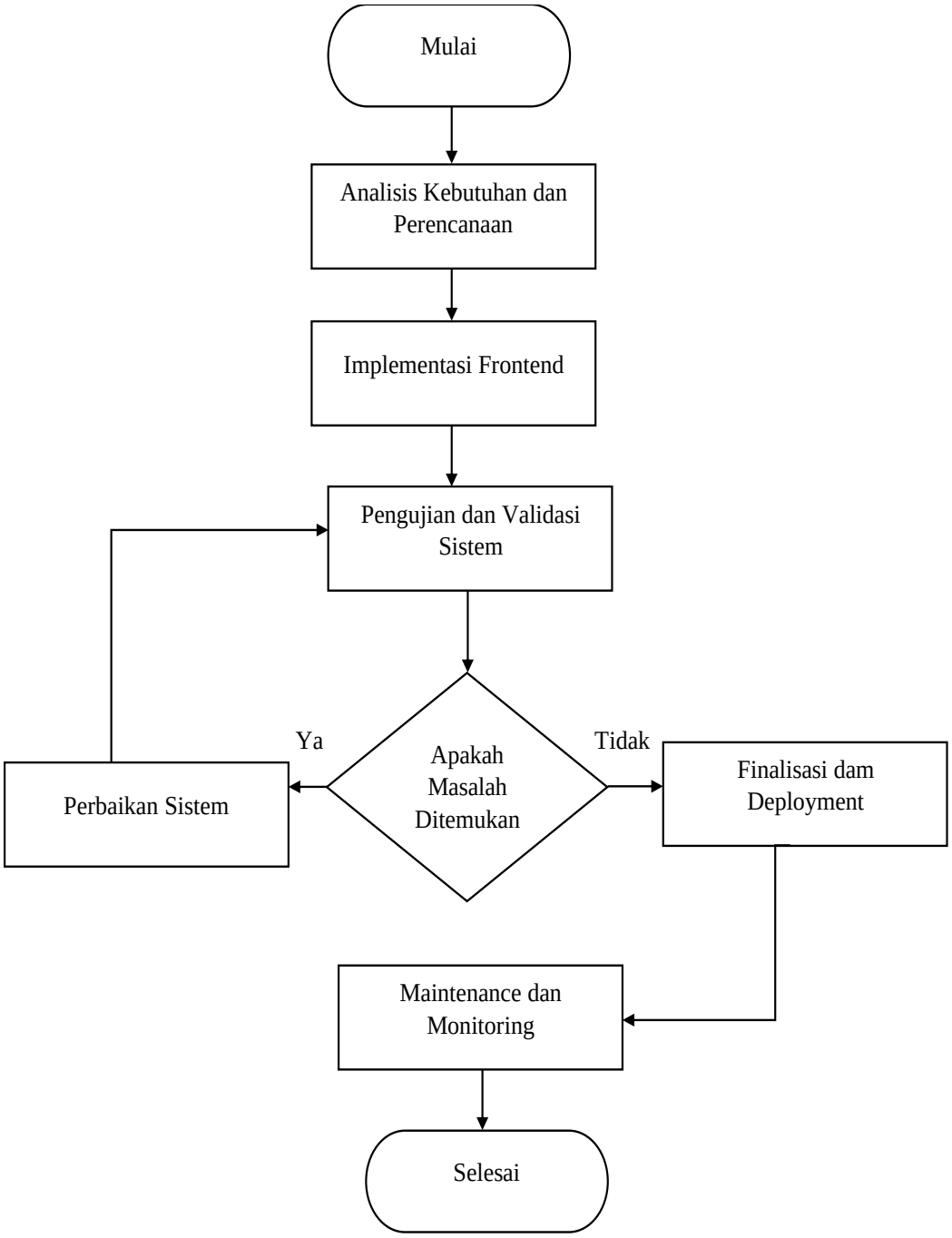
Kebutuhan fungsional meliputi pemantauan data lingkungan secara *real-time* dari sensor *IoT* dan perangkat *rover*, seperti kelembapan tanah, suhu udara, serta kondisi vegetasi. Sistem juga harus mendukung akses lintas *platform* melalui aplikasi *mobile* dan memiliki fitur *offline-first*, sehingga data tetap tersimpan dan tersinkronisasi otomatis saat koneksi internet tersedia kembali.

Sistem harus mampu mengolah data visual dari rover dan sensor rover secara efisien dengan antarmuka yang intuitif, responsif, serta mudah digunakan. Pengguna juga memerlukan fitur kustomisasi data agar dapat menampilkan parameter lingkungan sesuai kebutuhan operasional. *Server* yang andal diperlukan untuk menjamin sinkronisasi data secara berkala.

Pendekatan tersebut membuat sistem tidak hanya memenuhi kebutuhan operasional saat ini, tetapi juga mampu beradaptasi terhadap perkembangan teknologi dan dinamika lapangan di masa mendatang.

3.5. Pengembangan sistem

Metode Kanban mempresentasikan setiap tahap pekerjaan direpresentasikan secara visual guna mengoptimalkan efisiensi serta keterlacakan proses pengembangan. Kanban merupakan pendekatan manajemen alur kerja yang menekankan pada visualisasi tugas secara berkelanjutan, sehingga seluruh anggota tim dapat memantau status pekerjaan secara *real-time*.



Gambar 3.2 Tahap pengembangan sistem dengan metode kanban

Gambar 3.2 ini menggambarkan siklus hidup pengembangan sistem yang dimulai dari tahap perencanaan hingga pemeliharaan. Tantangan utama dalam proses ini adalah memastikan bahwa sistem yang dikembangkan bebas dari kesalahan dan siap digunakan. Tanpa alur yang terstruktur untuk pengujian dan perbaikan, ada

risiko besar merilis produk yang bermasalah, yang dapat merusak pengalaman pengguna dan menurunkan kredibilitas sistem.

Sistem dikembangkan melalui serangkaian tahapan yang jelas untuk mengatasi permasalahan tersebut. Proses pengembangan diawali dengan analisis kebutuhan dan implementasi *frontend*, yang kemudian diikuti oleh tahap pengujian dan validasi secara menyeluruh. Titik kunci alur ini terletak pada keputusan setelah pengujian: jika masalah ditemukan, sistem akan kembali ke tahap perbaikan sebelum diuji kembali untuk memastikan bahwa setiap bug berhasil diatasi. Proses pengembangan hanya dapat dilanjutkan ke tahap finalisasi dan *deployment* apabila tidak ditemukan masalah.

Pendekatan berbasis siklus ini membuktikan bahwa pengembangan sistem tidak berhenti saat produk dirilis. Sistem memasuki fase *maintenance* dan *monitoring* setelah *deployment* untuk menjamin performa serta kestabilannya dalam jangka panjang. Alur kerja ini memastikan bahwa hanya sistem yang berkualitas tinggi dan andal yang sampai ke tangan pengguna, sekaligus menyiapkan fondasi untuk perawatan berkelanjutan.

3.5.1 Tahap *Task To Do*

Tahap ini mencatat dan mengidentifikasi seluruh tugas yang berkaitan dengan pengembangan sistem secara sistematis. Lingkup pekerjaan mencakup perancangan antarmuka pengguna serta pengembangan fitur pemantauan drone berbasis *Progressive Web App (PWA)* menggunakan *React.js*. Setiap tugas disusun berdasarkan tingkat prioritas agar tim pengembang memiliki pemahaman yang jelas dan terarah mengenai ruang lingkup proyek sebelum proses implementasi dimulai.

3.5.2 Tahap *work in Progres*

Setelah seluruh tugas teridentifikasi, proses pengembangan sistem memasuki tahap *Work in Progress* dengan fokus pada implementasi modul utama, integrasi sensor *drone* untuk pengumpulan data, serta penerapan arsitektur *frontend* berbasis

React.js. Kolaborasi antara pengembang, desainer *UI/UX*, dan tim teknis *drone* menjadi faktor penting agar sistem berfungsi sesuai spesifikasi.

Kemajuan proyek dipantau melalui sistem manajemen proyek untuk memastikan kesesuaian antara hasil dan rencana kerja. Pengujian awal terhadap setiap modul dilakukan secara paralel untuk mengantisipasi potensi konflik. Komunikasi antar tim dijaga melalui pembaruan status dan pertemuan rutin, sehingga proses pengembangan tetap konsisten, efisien, dan tepat waktu.

3.5.3 Tahap *Testing*

Tahap *testing* bertujuan memastikan setiap fungsi sistem berjalan sesuai dengan kebutuhan dan standar performa yang telah ditetapkan. Tiga metode utama digunakan, yaitu unit *testing* untuk memeriksa fungsi setiap komponen, *integration testing* untuk memastikan keselarasan antar modul, serta *user acceptance testing (UAT)* guna menilai kesesuaian sistem dengan kebutuhan pengguna dalam pemantauan perkebunan sawit secara *real-time* menggunakan *rover*.

Tim pengembang juga melakukan *black-box testing* untuk menguji fungsionalitas sistem berdasarkan masukan (*input*) dan keluaran (*output*). Tim pengembang selanjutnya melaksanakan *usability testing* untuk menilai kemudahan penggunaan antarmuka. Tim pengembang kemudian mengadakan *performance testing* untuk mengukur efisiensi, kecepatan respons, dan kestabilan sistem saat memproses data lapangan secara berkelanjutan.

3.5.3.1 Pengujian Performa Menggunakan *PageSpeed Insights*

Pengujian performa *frontend* dilakukan dengan menggunakan *Google PageSpeed Insights (PSI)*. Tujuannya adalah menilai efisiensi waktu muat halaman serta kualitas pengalaman pengguna berdasarkan metrik standar, seperti *First Contentful Paint (FCP), Largest Contentful Paint (LCP), Total Blocking Time (TBT),* dan *Cumulative Layout Shift (CLS)*.

Pengujian dilaksanakan dalam konteks penggunaan nyata (*real-world context*) agar hasilnya mencerminkan pengalaman pengguna akhir. Evaluasi ini juga memungkinkan perbandingan performa antarversi aplikasi serta membantu mengidentifikasi hambatan pada aspek visual maupun interaktif. Dengan demikian, *PageSpeed Insights* berperan sebagai alat diagnostik sekaligus strategis untuk meningkatkan performa antarmuka, terutama pada kondisi jaringan terbatas atau perangkat dengan spesifikasi menengah.

3.5.3.2 *Blackbox Testing*

Black-box testing merupakan metode pengujian yang menilai fungsi sistem berdasarkan keluaran tanpa melihat struktur internal aplikasi. Teknik ini menggunakan pendekatan seperti *equivalence partitioning, boundary value analysis*, dan *state transition testing* untuk memastikan keandalan serta validitas hasil sistem.

Metode ini efektif dalam mendeteksi kesalahan pada antarmuka, logika proses, dan respons terhadap berbagai masukan pengguna. Pendekatan ini sangat relevan bagi aplikasi berbasis *React.js* dan *Progressive Web App (PWA)* karena berfokus pada interaksi pengguna serta validasi hasil tampilan. Selain itu, pengujian juga mencakup verifikasi terhadap fungsionalitas *offline-first* melalui *Service Worker*.

3.5.3.3 Performance *Testing*

Performance testing bertujuan mengukur kecepatan akses serta efisiensi sistem dalam kondisi penggunaan nyata oleh pengguna akhir. Pengujian ini menekankan evaluasi terhadap waktu muat, respons antarmuka, dan kestabilan aplikasi ketika diakses melalui berbagai perangkat.

Penelitian ini menggunakan *Google PageSpeed Insights* sebagai alat utama untuk menilai performa halaman pada perangkat mobile maupun desktop. Evaluasi penelitian mencakup sejumlah metrik penting, seperti *First Contentful Paint* (*FCP*),

Largest Contentful Paint (*LCP*), dan *Total Blocking Time* (*TBT*) sebagai indikator performa keseluruhan [33].

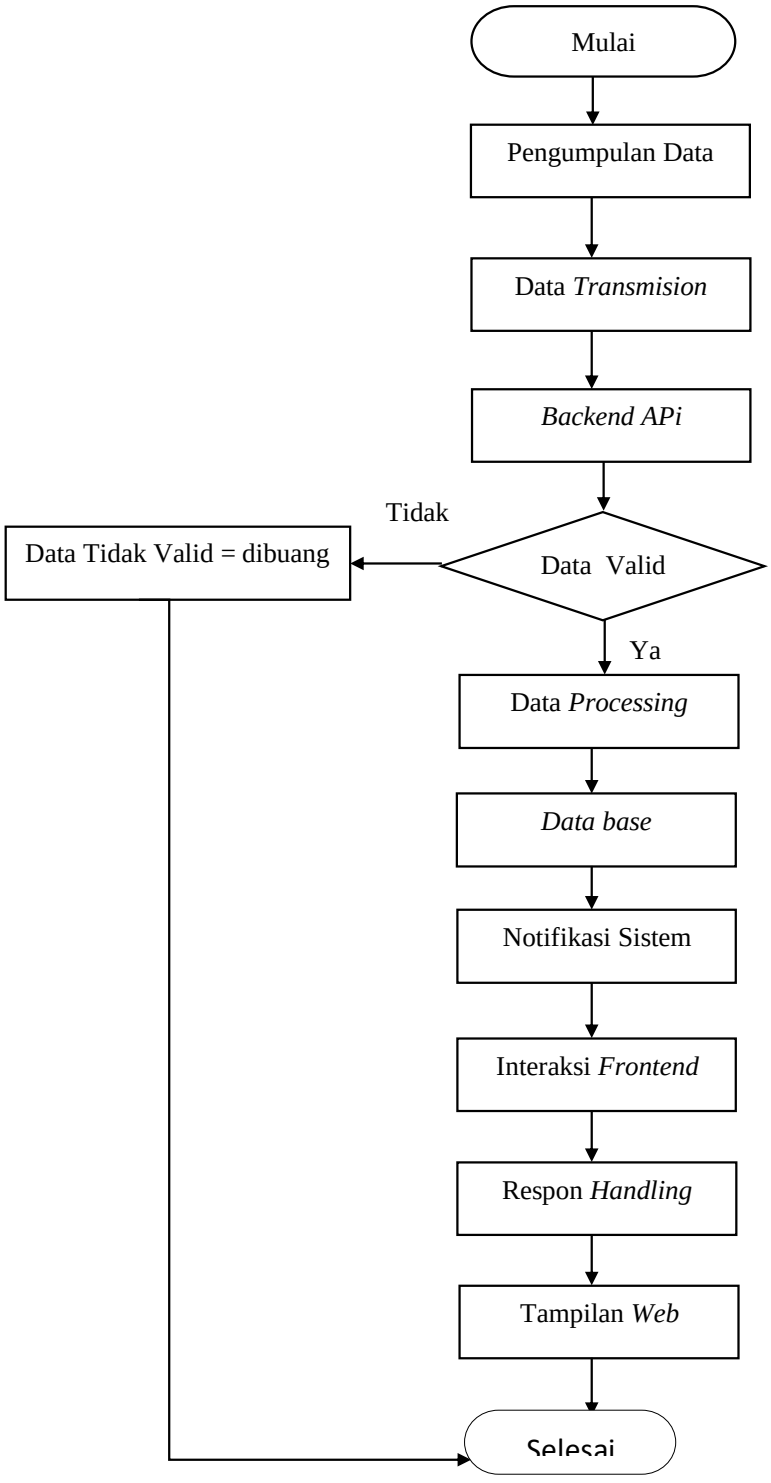
3.5.4 Tahap *Done*

Tahap ini menandai penyelesaian seluruh proses pengembangan sistem. Sistem *drone rover* yang dibangun menggunakan *React.js* dan *Progressive Web App (PWA)* berhasil di-*deploy* ke lingkungan produksi. Seluruh dokumentasi teknis, seperti panduan instalasi, konfigurasi sistem, dan petunjuk penggunaan, disusun secara lengkap sebagai referensi untuk pemeliharaan lanjutan.

Metode Kanban digunakan dalam pengelolaan proyek karena dinilai lebih fleksibel terhadap perubahan dan lebih efisien dibandingkan pendekatan tradisional seperti *Scrum* atau *Waterfall*. *Kanban* memungkinkan visualisasi alur kerja secara jelas melalui papan *Kanban* yang terdiri atas empat kolom, yaitu *Task To Do, Work In Progress*, *Testing*, dan *Done*. Visualisasi ini mempermudah pengelolaan tugas, membantu tim dalam memantau status pekerjaan, serta mendukung pengambilan keputusan berbasis prioritas.

3.6 Alur Data

Alur data sistem ditunjukkan melalui *flowchart* yang menggambarkan proses dari pengumpulan hingga penyajian informasi pada antarmuka pengguna. Proses diawali dengan pengumpulan data oleh perangkat, kemudian data diproses dan disimpan dalam sistem. Selain itu, sistem secara otomatis memperbarui data secara berkala agar informasi yang ditampilkan tetap akurat.



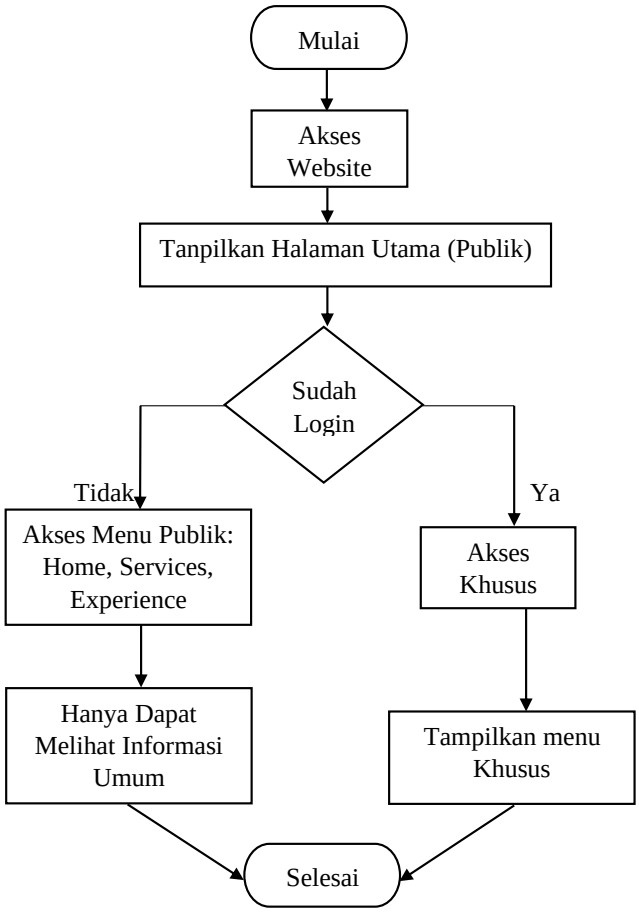
Gambar 3.3 Alur Data Sensor

Gambar 3.3 memaparkan aliran data dari tahap pengumpulan hingga penyajian informasi kepada pengguna. Proses ini berawal dari pengambilan data lingkungan yang dilakukan oleh perangkat *Internet of Things (IoT)*, seperti sensor suhu,

kelembapan tanah, dan intensitas cahaya. Setelah data tersebut diterima oleh broker, informasi ini kemudian diteruskan ke *Backend API* untuk menjalani proses lebih lanjut. Tahap pemrosesan di backend mencakup validasi format, tipe data, serta nilai numerik. Apabila data tersebut dinyatakan valid, sistem akan melanjutkannya ke tahap pemrosesan berdasarkan logika bisnis yang telah ditentukan sebelumnya.

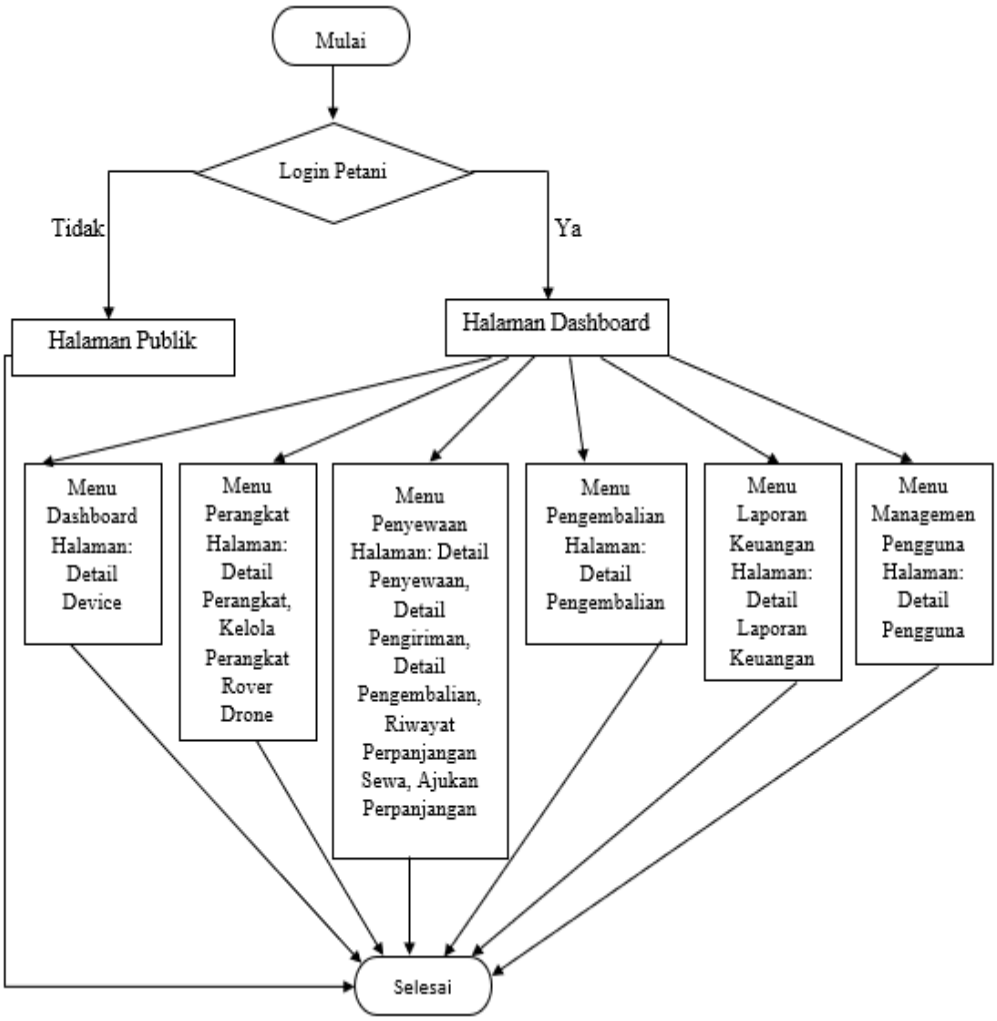
3.7 Tiga Parameter Pengguna dan Sistem *Role* nya

Sistem ini melibatkan tiga jenis pengguna utama, yaitu pengguna umum, petani, serta admin/operator sistem. Berikut adalah gambaran akses sistem yang bisa dilakukan oleh pengguna umum.



Gambar 3.4 Diagram Akses Sistem umum

Berdasarkan Gambar 3.4 yang menampilkan Diagram Sistem *User* Umum, pengguna dengan peran petani sebenarnya memiliki dua tingkatan akses. Sebelum melakukan *login*, pengguna hanya dapat melihat halaman publik seperti *Home*, *Services*, dan *Experience* yang berisi informasi umum. Setelah menjalani proses login, petani akan memperoleh akses penuh ke *Dashboard*, termasuk menu pengelolaan perangkat, penyewaan, pengiriman, serta berbagai transaksi.



Gambar 3.5 Diagram Sistem *user* Petani

Gambar 3.5 menggambarkan alur akses untuk pengguna dengan peran Petani di dalam sistem *Rover Hub*. Diagram ini memetakan perjalanan pengguna ke dalam dua wilayah akses yang berbeda. Proses *login* menjadi penentu utama yang memisahkan kedua wilayah tersebut secara tegas. Proses *login* ini berfungsi sebagai gerbang utama yang menentukan sejauh mana pengguna dapat berinteraksi dengan

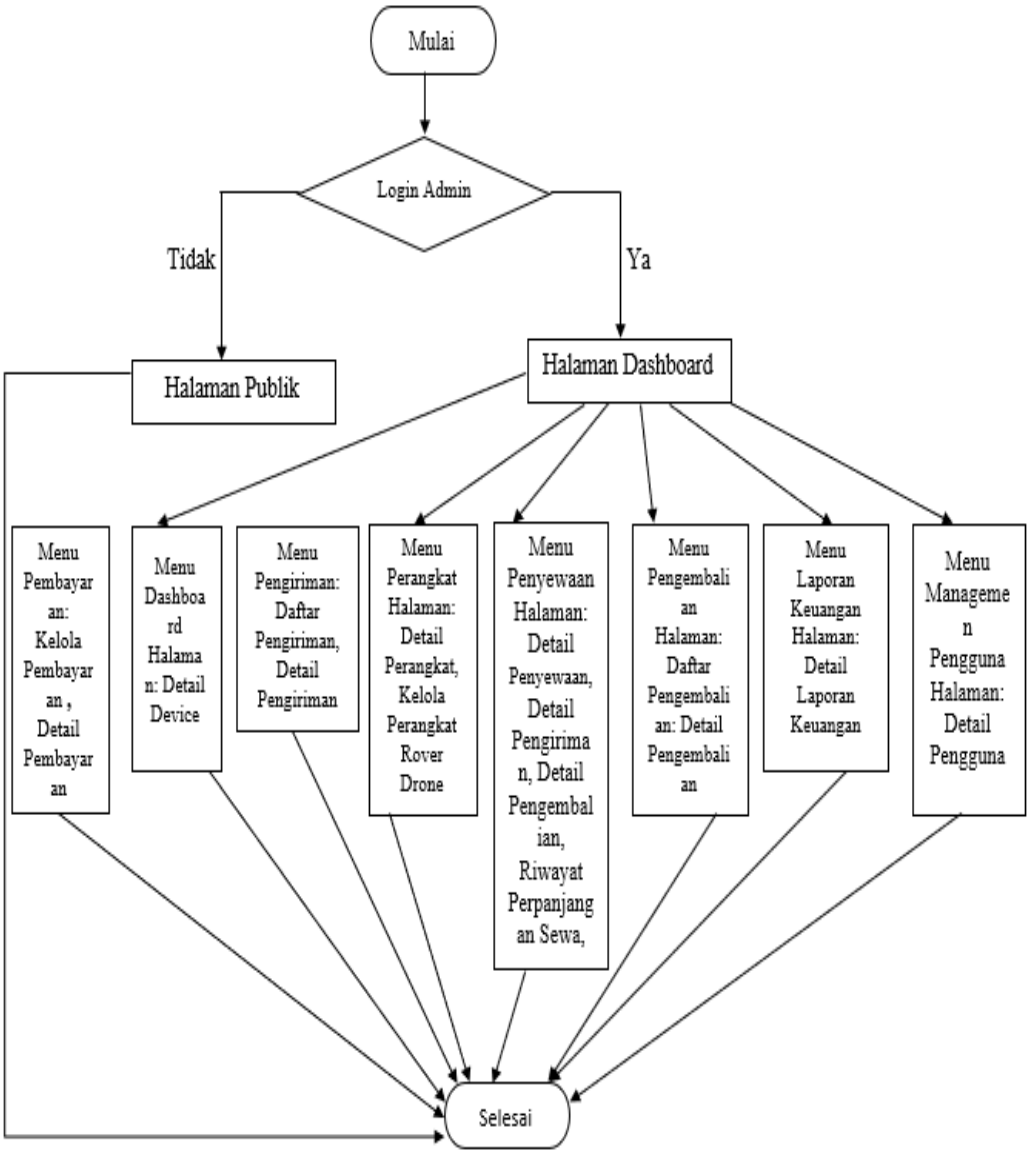
fitur-fitur *platform*, dan karena itu, sistem ini menjaga keamanan data dan fungsionalitas.

Sistem membatasi akses Petani sehingga sebelum login mereka hanya dapat membuka halaman publik. Platform menyajikan informasi umum melalui Menu Home, Services, dan Experience untuk memberikan gambaran awal kepada calon pengguna. Informasi tersebut berfungsi sebagai orientasi awal mengenai fitur platform yang tersedia. Kondisi ini membuat Petani belum dapat melakukan tugas-tugas operasional, sehingga terdapat pemisahan yang jelas antara pengunjung dan pengguna yang sudah terdaftar.

Petani akan masuk ke Halaman Dashboard setelah login, yang berfungsi sebagai pusat kendali pribadi. Halaman Dashboard menjadi pintu gerbang menuju berbagai menu utama yang disusun secara horizontal. Menu utama tersebut mencakup Menu Dashboard, Menu Perangkat, Menu Penyewaan, Menu Pengembalian, dan Menu Laporan Keuangan.

Setiap menu utama menyediakan akses ke halaman spesifik yang memungkinkan Petani melakukan tugas operasional secara mandiri. Melalui Menu Perangkat, Petani dapat melihat daftar unit dan membuka *Detail Perangkat* untuk memperoleh informasi teknis. Melalui Menu Penyewaan, Petani dapat melihat *Detail Penyewaan*, mengajukan perpanjangan sewa, serta memantau status pengiriman dan pengembalian, sehingga pengguna memperoleh pemberdayaan penuh.

Diagram ini menunjukkan filosofi desain sistem yang memisahkan ruang informasi publik dan ruang kerja privat. Struktur alur tersebut tidak hanya menjaga keamanan, tetapi juga membentuk pengalaman pengguna yang berorientasi pada tugas. Sistem memberikan kemampuan bagi Petani untuk mengelola seluruh siklus penyewaan secara mandiri setelah login, mulai dari pemilihan perangkat, pemantauan, hingga transaksi keuangan. Pada akhirnya, rancangan sistem ini mendukung efisiensi dan kemandirian pengguna.



Gambar 3.6 Diagram Sistem *admin*

Gambar 3.6 menunjukkan alur akses penuh yang dimiliki Admin dalam sistem *Rover Hub*, yang dimulai dari proses autentikasi *login* sebagai gerbang utama. Diagram ini menggambarkan dua kemungkinan hasil: kegagalan *login* akan mengarahkan pengguna kembali ke halaman publik, sedangkan keberhasilan *login* akan membawa *Admin* masuk ke *Dashboard* yang berfungsi sebagai pusat operasional.

Admin mengakses menu utama di dalam Dashboard yang terbagi ke dalam tiga kategori besar. Sistem menyediakan Manajemen Sumber Daya untuk mengelola

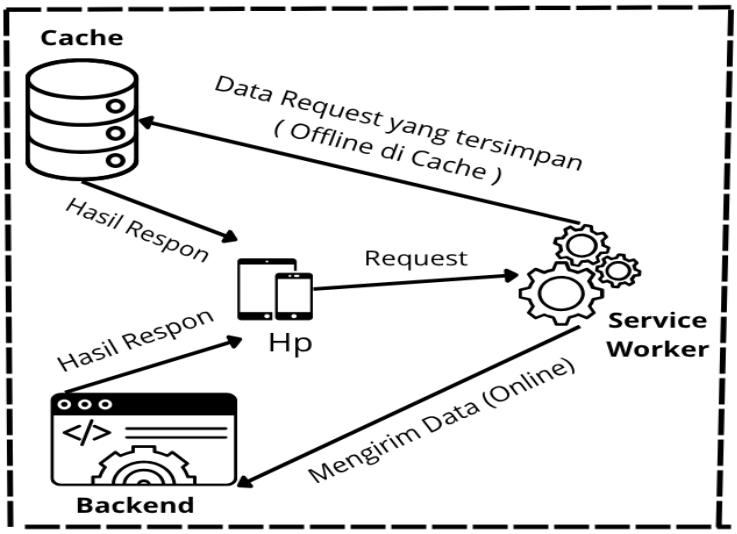
data pengguna dan inventaris perangkat. Platform juga menghadirkan Manajemen Operasional yang bertugas mengawasi seluruh siklus layanan mulai dari penyewaan hingga pengembalian. Sistem tambahan berupa Manajemen Informasi berfungsi sebagai pusat data analitik.

Setiap menu memberikan aksi spesifik yang menegaskan wewenang penuh Admin. Pada Manajemen Operasional, Admin dapat menyetujui atau menolak permohonan penyewaan. Pada Manajemen Informasi, Admin dapat menganalisis keuangan dan memantau aktivitas sistem melalui laporan yang tersedia sebagai dasar pengambilan keputusan strategis.

Diagram ini menunjukkan peran Admin sebagai supervisor tertinggi dengan kendali dan tanggung jawab yang menyeluruh. Struktur alur memastikan seluruh proses bisnis, mulai dari manajemen aset hingga logistik, dapat dipantau dan dievaluasi secara efektif, sehingga integritas dan kelancaran platform tetap terjaga.

3.8 Penerapan *Progressive Web App* (PWA) Pada Sistem *Front end*

Dalam pengembangan antarmuka pengguna berbasis *React.js*, teknologi *Progressive Web App (PWA)* diimplementasikan guna meningkatkan performa, keandalan, dan ketersediaan layanan, terutama pada kondisi jaringan yang terbatas. Teknologi *PWA* memungkinkan aplikasi untuk berjalan layaknya sebuah aplikasi native. Fitur-fitur seperti mode *offline*, kemampuan instalasi pada perangkat, serta *caching* aset statis melalui *Service Worker* menjamin sistem tetap dapat digunakan bahkan tanpa adanya koneksi internet.

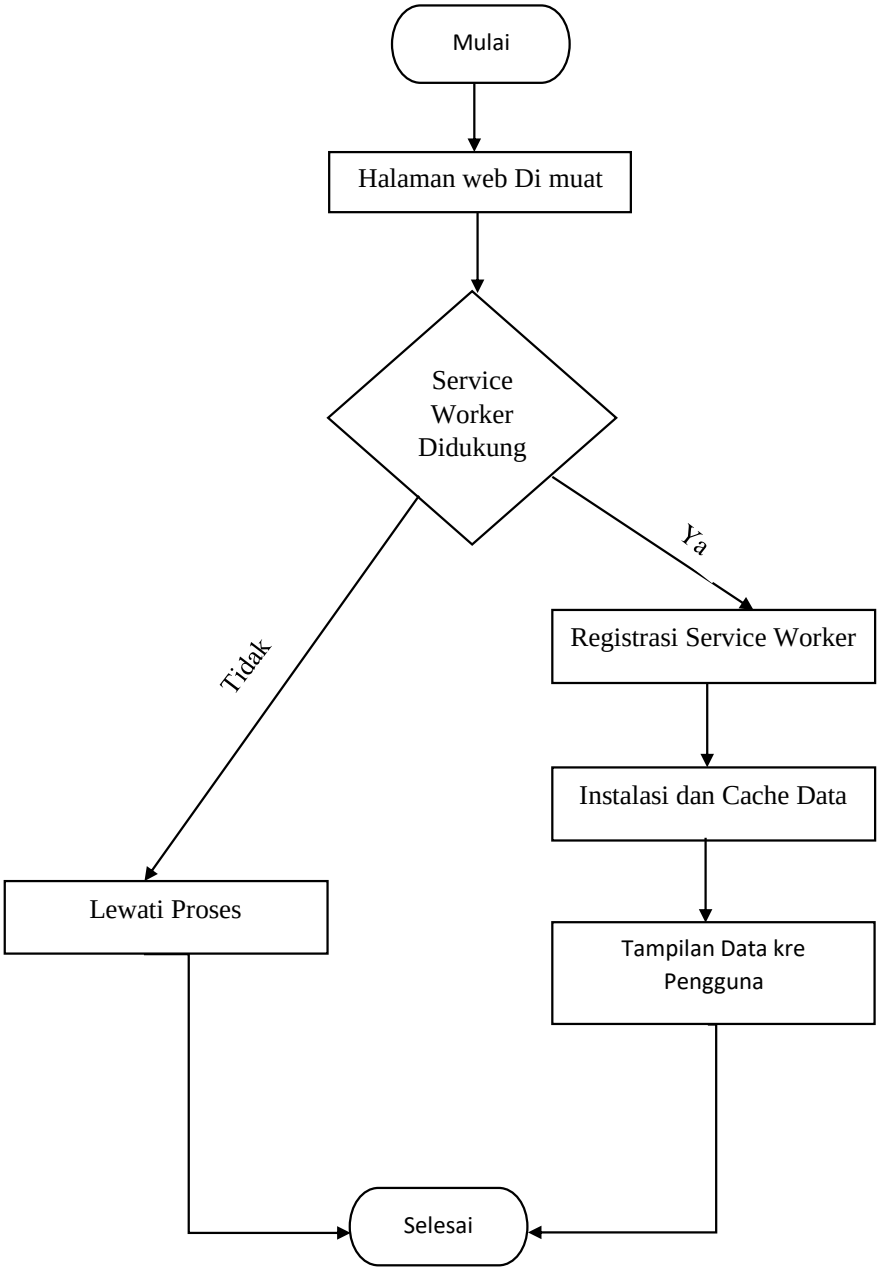


Gambar 3.7 Penerapan *Progressive Web App* (PWA) Pada Sistem *Front end*

Gambar 3.7 memaparkan mekanisme kerja dari *Service Worker* dalam mengatur komunikasi antara *frontend* dan *backend* pada sistem *Rover Drone* berbasis *Progressive Web App (PWA)*. Proses ini menjelaskan bagaimana aplikasi mampu tetap berfungsi dengan baik meskipun tidak terhubung ke internet.

Pengguna melalui perangkat *mobile* (HP) mengajukan sebuah permintaan, permintaan tersebut akan diteruskan kepada *Service Worker* untuk diperiksa terlebih dahulu. Jika data yang diminta sudah tersimpan di *cache*, maka hasil respons akan langsung diambil dari sana tanpa harus mengakses jaringan. Hal inilah yang memungkinkan aplikasi untuk tetap berjalan dalam kondisi *offline*. Akan tetapi, apabila data belum tersimpan atau pengguna sedang dalam kondisi *online*, *Service Worker* akan mengirimkan permintaan tersebut ke backend untuk memperoleh data yang paling baru.

Alur ini secara jelas mencerminkan penerapan konsep *offline-first*, di mana *Service Worker* bertindak sebagai penghubung antara data lokal dan server. Dengan pendekatan ini, aplikasi dapat berjalan dengan cepat, hemat bandwidth, serta tetap responsif baik dalam keadaan *online* maupun *offline*, yang merupakan prinsip utama dari sebuah *PWA*.



Gambar 3.8 Alur Pendaftaran dan Mekanisme Operasional *Service Worker*

Gambar 3.8 menggambarkan mekanisme kerja *Service Worker* pada sistem aplikasi *Rover Drone* berbasis *Progressive Web App* (PWA). Proses dimulai ketika pengguna mengakses halaman *web* aplikasi. Setelah halaman berhasil dimuat, sistem melakukan pemeriksaan terhadap dukungan *Service Worker* pada peramban (*browser*) yang digunakan.

Apabila *Service Worker* didukung oleh peramban, maka sistem melanjutkan proses ke tahap registrasi *Service Worker*. Setelah proses registrasi berhasil, dilakukan instalasi serta penyimpanan (*caching*) berbagai data penting seperti berkas *HTML*, *CSS*, *JavaScript*, serta aset-aset pendukung lainnya yang diperlukan untuk meningkatkan performa aplikasi. Selanjutnya, data hasil pemrosesan akan disajikan kepada pengguna sesuai dengan alur kerja yang diatur oleh *Service Worker*.

Sistem akan melewati tahap registrasi dan instalasi apabila peramban tidak mendukung *Service Worker*. Kondisi tersebut membuat aplikasi menampilkan data secara langsung tanpa memanfaatkan mekanisme penyimpanan sementara (*cache*) yang biasanya dikelola oleh *Service Worker*.

Flowchart pada gambar ini menjelaskan secara sistematis bagaimana sistem memeriksa dukungan terhadap *Service Worker*, melaksanakan proses registrasi dan instalasi ketika dukungan tersedia, serta memastikan bahwa data tetap dapat diakses oleh pengguna meskipun tidak ada dukungan penuh terhadap *Service Worker*.

3.9 Analisa Hasil

Tahap analisis hasil bertujuan mengevaluasi kinerja dan fungsionalitas antarmuka pengguna pada sistem pemantauan dan pemeliharaan perkebunan sawit berbasis *Progressive Web App (PWA)* menggunakan *React.js.* Evaluasi difokuskan pada kecepatan respons, kompatibilitas antarperangkat, serta efektivitas sinkronisasi data pada kondisi daring dan luring.

Pengujian dilakukan menggunakan alat analisis seperti *Google PageSpeed Insights* untuk memantau waktu muat halaman dan kestabilan tampilan. Hasil pengujian disajikan melalui *dashboard* interaktif yang diakses melalui *URL PWA* yang di*hosting* pada *Virtual Private Server (VPS).*

3.10 Alasan Pemilihan *React.js* Dibanding *Framework* Lain

Pemilihan *React.js* didasarkan pada pertimbangan teknis dan fungsional yang sesuai dengan kebutuhan sistem. *React.js* memiliki arsitektur berbasis komponen yang mendukung pengembangan modular, mudah dipelihara, dan bersifat skalabel.

React.js memiliki ekosistem yang luas serta komunitas aktif yang mendukung integrasi dengan *Progressive Web App (PWA)*. Teknologi *Virtual DOM* memberikan keunggulan dalam melakukan pembaruan data secara *real-time* dengan performa yang stabil, suatu aspek penting dalam proses pemantauan data sensor drone–rover.

Framework lain seperti *Vue.js* memang menawarkan ukuran yang lebih ringan, tetapi ekosistem dan dukungan pengembangannya masih terbatas. Oleh sebab itu, React.js menjadi pilihan paling tepat untuk penelitian ini.

3.11 Etika Penelitian dan Batasan Tambahan

Etika penelitian dijaga untuk memastikan bahwa seluruh proses pengembangan sistem tidak menimbulkan dampak negatif terhadap lingkungan, keamanan data, maupun kepentingan pengguna. Data yang digunakan sebagian besar berupa data simulasi (*dummy* data), kecuali pengujian terbatas yang menggunakan data nyata untuk validasi awal.

Penelitian ini juga menetapkan batasan bahwa konektivitas internet di wilayah perkebunan bersifat fluktuatif, bukan sepenuhnya tidak tersedia. Oleh karena itu, fitur offline-first dirancang untuk mengatasi gangguan konektivitas jangka pendek. Penelitian ini tidak mencakup kondisi blank spot total tanpa infrastruktur jaringan.

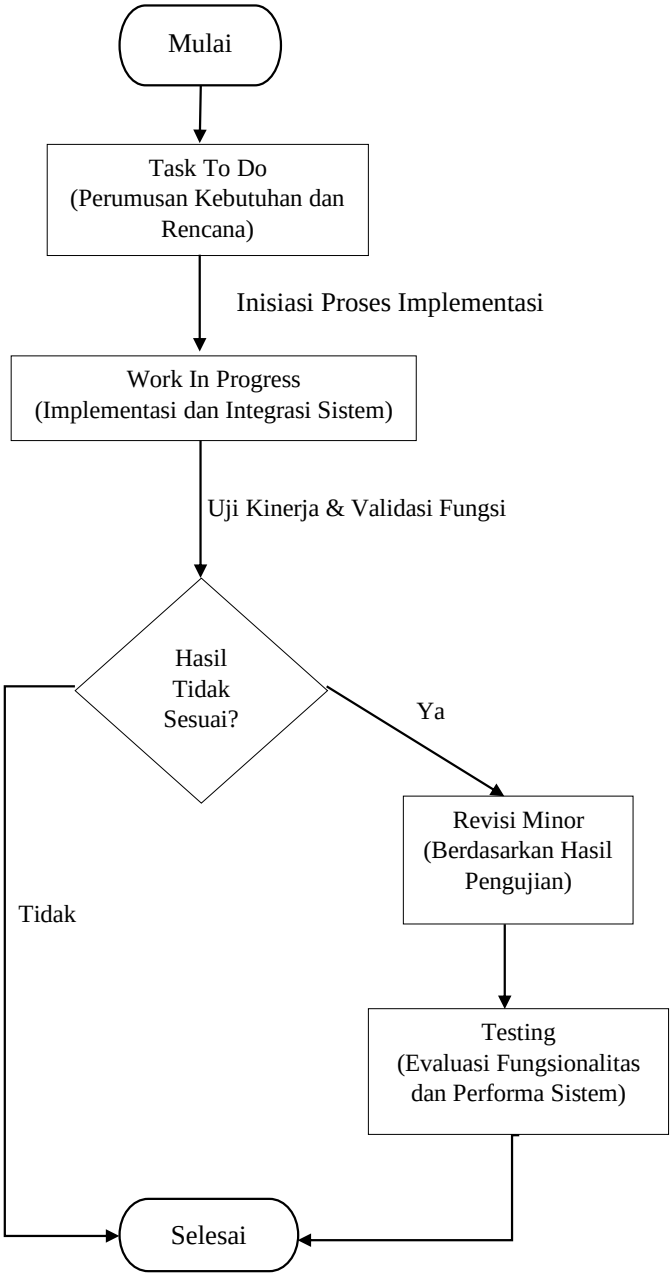
3.12 Metode Pengembangan dengan Pendekatan Kanban

Metode pengembangan sistem ini menggunakan pendekatan Kanban, yaitu metode manajemen kerja dalam kerangka *Agile* yang menekankan visualisasi alur proses, pembatasan pekerjaan aktif (*Work In Progress*), dan prinsip *continuous improvement*. Pendekatan ini sesuai dengan karakter pengembangan *front-end*

Progressive Web App (PWA) berbasis *React.js* yang menuntut fleksibilitas tinggi dan pemantauan progres secara transparan.

Proses *Kanban* terdiri atas empat tahap utama: *Task To Do*, *Work In Progress*, *Testing*, dan *Done*. Tahap *Task To Do* berfokus pada perumusan kebutuhan dan perencanaan tugas, sedangkan *Work In Progress* meliputi implementasi, integrasi *API*, serta konfigurasi fitur *PWA* dengan pembatasan tugas agar efisien. Tahap *Testing* dilakukan untuk memverifikasi hasil menggunakan *Google PageSpeed Insights*, *Blackbox Testing*, dan *Performance Testing*. Komponen yang telah lolos pengujian kemudian dipindahkan ke tahap *Done* untuk integrasi akhir dan dokumentasi.

Secara keseluruhan, pendekatan *Kanban* memberikan proses pengembangan yang terstruktur, efisien, dan adaptif, sekaligus menjamin kualitas sistem melalui evaluasi dan pengendalian berkelanjutan..



Gambar 3.9 Diagram Alur Penerapan Metode Kanban dalam Pengembangan Sistem

Gambar 3.9 menggambarkan alur penerapan metode Kanban dalam proses pengembangan sistem yang terdiri dari empat tahapan utama: *Task To Do*, *Work In Progress*, *Testing*, dan *Done*. *Tahap Task To Do* mencakup perumusan kebutuhan dan perencanaan pengembangan, sedangkan *Work In Progress* merupakan tahap implementasi dan integrasi sistem sebagai inti dari proses pengembangan. Setelah itu, tahap *Testing* dilakukan untuk mengevaluasi fungsionalitas dan performa

sistem agar sesuai dengan kebutuhan yang telah ditetapkan, sebelum akhirnya hasil akhir diselesaikan dan divalidasi pada tahap *Done*.

Panah putus-putus dari *Testing* menuju *Work In Progress* menunjukkan adanya revisi minor berdasarkan hasil pengujian, yang mencerminkan prinsip *continuous improvement* dalam metode Kanban. Secara keseluruhan, diagram ini menampilkan alur kerja yang terstruktur, efisien, dan adaptif terhadap perubahan, sehingga mendukung pengembangan sistem yang berkelanjutan dan berkualitas.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian berjudul "Pengembangan *Front-End* Sistem *Rover Drone Otonom* Berbasis *PWA* Menggunakan *React.js* untuk Pemantauan dan Pemeliharaan Perkebunan Sawit", dapat disimpulkan beberapa hal sebagai berikut.

1. Sistem antarmuka pengguna berbasis web berhasil dikembangkan menggunakan *React.js* dan diimplementasikan dalam bentuk *Progressive Web App (PWA)*. Sistem ini mendukung lintas *platform* pada perangkat *desktop* maupun *mobile*, sesuai dengan tujuan penelitian untuk menghadirkan solusi pemantauan perkebunan yang fleksibel, adaptif, serta mudah diakses oleh pengguna.
2. Penerapan *PWA* mencakup komponen inti seperti *service worker*, *manifest.json*, *cache offline*, fitur instalasi, *splash screen*, dan mode *standalone*. Hasil pengujian *black-box* dengan 12 indikator performa *PWA* menunjukkan tingkat kelayakan yang baik serta memenuhi *best practices* dalam pengembangan aplikasi modern.
3. Sistem mampu menampilkan dan mengelola data secara waktu nyata (*real-time*) dari sensor *drone rover*, meliputi parameter suhu udara, kelembapan tanah, dan intensitas pencahayaan. Kemampuan ini mendukung proses pemantauan kondisi lingkungan perkebunan secara akurat dan berkelanjutan.
4. Penerapan fitur *offline-first* pada arsitektur *PWA* terbukti efektif dalam menjaga ketersediaan layanan meskipun konektivitas internet tidak stabil.

5. Fitur tersebut sangat relevan diterapkan di wilayah perkebunan kelapa sawit yang umumnya berada di daerah terpencil dengan keterbatasan infrastruktur jaringan.
6. Hasil evaluasi performa menunjukkan skor keseluruhan *PWA* sebesar 80%. Nilai ini menandakan bahwa sebagian besar aspek *installability*, kapabilitas *offline*, serta efisiensi akses telah terpenuhi dengan baik, sehingga aplikasi layak digunakan sebagai sistem pemantauan berbasis web yang andal.

5.2 Saran

Untuk pengembangan lebih lanjut, beberapa saran yang dapat dipertimbangkan antara lain:

1. Optimalisasi performa aplikasi melalui teknik seperti *lazy loading*, *code splitting*, dan kompresi ukuran file agar waktu pemuatan lebih cepat serta efisiensi penggunaan sumber daya meningkat.
2. Penambahan fitur notifikasi dan sinkronisasi latar belakang (*push notification* dan *background sync*) guna memberikan informasi terbaru kepada pengguna tanpa memerlukan pembaruan manual.
3. Pengujian di lapangan nyata pada wilayah perkebunan yang berbeda untuk memvalidasi kinerja *offline-first* dan memastikan sistem dapat berfungsi optimal di berbagai skenario jaringan.
4. Peningkatan keamanan data melalui enkripsi data sensor dan mekanisme autentikasi yang lebih ketat untuk melindungi informasi sensitif dari potensi risiko keamanan.
5. Kolaborasi dengan industri kelapa sawit dalam skala luas guna mendukung integrasi sistem dengan teknologi *Internet of Things (IoT)*, pemrosesan data berbasis *Artificial Intelligence (AI)*, serta memastikan sistem dapat digunakan secara berkelanjutan di berbagai perkebunan.
6. Perluasan cakupan parameter pemantauan, misalnya mencakup pH tanah, kandungan unsur hara, atau deteksi dini serangan hama, sehingga sistem dapat memberikan analisis kondisi lahan yang lebih komprehensif.

DAFTAR PUSTAKA

[1] F. Isharyadi *et al.*, “Implementasi standar nasional Indonesia (SNI) pada produksi benih kelapa sawit di Indonesia,” *J. Penelit. Kelapa Sawit*, vol. 32, no. 1, pp. 57–70, Apr. 2024, doi: 10.22302/iopri.jur.jpks.v32i1.240.

[2] R. Chin *et al.*, “Plant disease detection using drones in precision agriculture,” *Precision Agriculture*, vol. 24, no. 5, pp. 1663–1682, Oct. 2023, doi: 10.1007/s11119-023-10014-y.

[3] D. Gao *et al.*, “A framework for agricultural pest and disease monitoring based on Internet-of-Things and unmanned aerial vehicles,” *Sensors*, vol. 20, no. 5, p. 1487, Mar. 2020, doi: 10.3390/s20051487.

[4] M. Gao *et al.*, “Automatic monitoring of maize seedling growth using unmanned aerial vehicle-based RGB imagery,” *Remote Sensing*, vol. 15, no. 14, p. 3671, Jul. 2023, doi: 10.3390/rs15143671.

[5] L. Sutiarso *et al.*, “Aplikasi sistem monitoring pertumbuhan tanaman berbasis web menggunakan machine vision,” *Agritech*, vol. 31, no. 4, pp. 359–367, Nov. 2021, doi: 10.22146/agritech.9644.

[6] S. Chen *et al.*, “Front-end development in React: an overview,” *Eng. Int.*, vol. 7, no. 2, pp. 117–126, Dec. 2019, doi: 10.18034/ei.v7i2.662.

[7] H.T. Wu, “Establish a digital real-time learning system with push notifications,” *Front. Psychol.*, vol. 13, p. 767389, Feb. 2022, doi: 10.3389/fpsyg.2022.767389.

[8] D. Shelar, et al., Kumbhar, “Automatic wall painting robot,” *Int. J. Innov. Eng. Res. Technol.* (IJIERT), vol. 5, no. 4, pp. 1–4, Apr. 2018, doi: 10.2394-3696/IJIERT.2018.12345.

[9] P. Koysawat *et al.*, “Progressive web app for crop field data collection,” in *Proc. IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1163, p. 012018, Jun. 2021, doi: 10.1088/1757-899X/1163/1/012018.

[10] R. Lavrenov *et al.*, “Graphical user interface design for a UAV teleoperation,” in *Proc. Int. Conf. Artif. Life Robot.* (ICAROB), Tokyo, Japan, Apr. 2022, pp. 678–681, doi: 10.5954/ICAROB.2022.OS17-2.

[11] K. Anderson and K. Gaston, “Lightweight unmanned aerial vehicles will revolutionize spatial ecology,” *Front. Ecol. Environ.*, vol. 11, no. 3, pp. 138–146, Mar. 2013, doi: 10.2307/23470549.

[12] F. M. J. M. Shamrat *et al.*, “A web-based application for agriculture: Smart farming system,” *Int. J. Emerg. Trends Eng. Res.*, vol. 8, no. 6, pp. 2309–2320, Jun. 2020, doi: 10.30534/ijeter/2020/18862020.

[13] E. López-Contreras *et al.*, “Web-interface for rover teleoperation to investigate the impact of extreme environments,” in *Proc. IAF Human Spaceflight Symp.*, Milan, Italy, Oct. 2024, pp. 820–826, doi: 10.52202/078364-0093.

[14] R. F. Malik *et al.*, “Real-time environmental monitoring in palm oil plantation using wireless sensor network,” in *Proc. Int. Conf. Electr. Eng., Comput. Sci. Informatics* (EECSI), Palembang, Indonesia, Aug. 2015, pp. 123–126, doi: 10.12345/EECSI.2015.67890.

[15] S. Boekle *et al.*, “Consideration of resilience for digital farming systems,” *Comput. Electron. Agric.*, vol. 190, p. 106456, Feb. 2021, doi: 10.1016/j.compag.2021.106456.

[16] N. S. Abu *et al.*, “Internet of Things applications in precision agriculture: a review,” *J. Robot. Control (JRC)*, vol. 3, no. 3, p. 14159, 2022, doi: 10.18196/jrc.v3i3.14159.

[17] B. Wang *et al.*, “Design and development of a local-first collaborative 3D WebGIS application for mapping,” *ISPRS Int. J. Geo-Inf.*, vol. 12, no. 5, pp. 123–145, May 2023, doi: 10.3390/ijgi12050234.

[18] M. S. S. Lingolu and M. K. Dobbala, “A comprehensive review of progressive web apps: Bridging the gap between web and native experiences,” *Int. J. Sci. Res.* (IJSR), vol. 11, no. 2, pp. 1325–1334, Feb. 2022, doi: 10.21275/SR24517172948.

[19] A. J. Thomas and S. R. Kumar, “A study on progressive web apps: Revolutionizing user experiences and redefining web applications,” *Int. J. Sci. Res.*, vol. 5, no. 6, Feb. 2024, doi: 10.29121/shodhkoshv5.i6.2024.5977.

[20] R. Fauzan *et al.*, “A systematic literature review on progressive web application practice and challenges,” *IPTEK J. Technol. Sci.*, vol. 33, no. 1, pp. 43–58, 2022, doi: 10.12962/j20882033.v33i1.13904.

[21] B. R. Cherukuri, “Progressive web apps (PWAs): Enhancing user experience through modern web development,” *Int. J. Sci. Res.* (IJSR), vol. 13, no. 10, pp. 1549–1560, Oct. 2024, doi: 10.21275/MS241022095359.

[22] A. Bhalla *et al.*, “Present day web development using ReactJS,” *Int. Res. J. Eng. Technol.* (IRJET), vol. 7, no. 5, pp. 1154–1157, May 2020, doi: 10.12345/IRJET.2020.54321.

[23] V. V. Veeri, “Performance optimization techniques in React applications: A comprehensive analysis,” *Int. J. Res. Comput. Appl. Inf. Technol.*, vol. 7, no. 2, pp. 1165–1177, Mar. 2024, doi: 10.5281/zenodo.14146734.

[24] A. Gallidabino and C. Pautasso, “Multi-device complementary view adaptation with liquid media queries,” *J. Web Eng.*, vol. 18, no. 8, pp. 761–800, Jan. 2020, doi: 10.13052/jwe1540-9589.1882.

[25] A. Setiawan and A. I. Purnamasari, “Implementasi JSON Web Token berbasis algoritma SHA-512 untuk otentikasi aplikasi BatikKita,” *J. RESTI*, vol. 4, no. 6, pp. 1036–1045, Dec. 2020, doi: 10.29207/resti.v4i6.2533.

[26] S. Jain *et al.*, “Building scalable IoT dashboards with MERN technologies,” *Int. J. Res. Publ. Rev.*, vol. 6, no. 5, pp. 7033–7037, May 2025, doi: 10.12345/IJRPR.2025.98765.

[27] E. Goyal *et al.*, “Implementation and comparison of MERN stack technology with HTML/CSS, SQL, PHP and MEAN in web development,” *Int. Res. J. Modern Eng. Technol. Sci.*, vol. 5, no. 11, pp. 45–60, Nov. 2023, doi: 10.56726/IRJMETS46315.

[28] I. Garg *et al.*, “Study on JSON, its uses and applications in engineering organizations,” *Res. Rep.*, Otto von Guericke Univ. Magdeburg, Mar. 2024, doi: 10.13140/RG.2.2.19850.07367.

[29] A. A. Joshi *et al.*, “Workflow management through Kanban: A review,” *Recent Trends Prod. Eng.*, vol. 7, no. 2, pp. 9–15, May 2024, doi: 10.5281/zenodo.11401600.

[30] E. Lin *et al.*, “UntrustIDE: Exploiting weaknesses in VS Code extensions,” in *Proc. Netw. Distrib. Syst. Security (NDSS) Symp.*, San Diego, USA, Feb. 2024, doi: 10.14722/ndss.2024.24073.

[31] C. Xilogianni *et al.*, “Speed matters: What to prioritize in optimization for faster websites,” *Analytics*, vol. 1, no. 2, pp. 175–192, Nov. 2022, doi: 10.3390/analytics1020012.

[32] D. Corradini *et al.*, “Empirical comparison of black-box test case generation tools for RESTful APIs,” *arXiv*, Aug. 2021, doi: 10.12345/arXiv.2021.98765.

[33] P. R. Saputro and R. F. A. Aziza, "PWA and non-PWA performance analysis: Chrome extension testing on e-commerce platform," *J. Ilmu Pengetahuan dan Teknol. Komput.*, vol. 10, no. 4, pp. 909–916, Jun. 2025, doi: 10.33480/jitk.v10i4.6238.

[34] K. I. Roumeliotis and N. D. Tselikas, "Evaluating progressive web app accessibility for people with disabilities," *Network*, vol. 2, no. 2, pp. 350–369, Jun. 2022, doi: 10.3390/network2020022.

[35] A. Muawwal, "The implementation of PWA (progressive web app) technology in enhancing website performance and mobile accessibility," *Buletin Pos dan Telekomunikasi*, vol. 22, no. 1, pp. 25–36, 2023, doi: 10.17933/bpostel.v22i1.395.