



Rancang Bangun Sistem Informasi Pengelolaan Sarana dan Prasarana Berbasis *Web* dengan Metode *Iterative Incremental*

Muhammad Rajab Fadillah¹, Joko Priambodo^{2*}, Muhammad Adha Kemal³, Muhammad Bagus Fajri⁴

^{1,2,3,4} Fakultas Ilmu Komputer, Program Studi Teknik Informatika, Universitas Pamulang, Tangerang Selatan, Indonesia

Email: ¹razabpasiron132@gmail.com, ^{2*}dosen00276@unpam.ac.id, ³m.adha.kemal71@gmail.com, ⁴xiuxchin@gmail.com

(* : corresponding author)

Abstrak—Penelitian ini mengembangkan SIPRAKAR, yaitu sistem informasi berbasis *web* untuk mendukung pengelolaan sarana dan prasarana pada Lembaga Pengelola Aset dan Urusan Dalam Universitas Pamulang. Masalah yang dikaji berangkat dari alur kerja yang masih tersebar pada dokumen terpisah, terutama pada pencatatan program kerja, penyusunan RAB, penugasan pekerjaan, pemantauan progres, dan penyusunan laporan. Kondisi tersebut menurunkan keterlacakan data dan memperlambat pengambilan keputusan operasional. Penelitian memakai metode *Iterative Incremental* melalui tahapan analisis kebutuhan, perancangan, implementasi bertahap, pengujian, dan penyempurnaan modul. Sistem dibangun dengan *Laravel*, *React*, *Inertia.js*, dan *MySQL*. Hasil pengembangan menunjukkan bahwa SIPRAKAR mampu menautkan program kerja, RAB, pekerjaan, *checklist progres*, laporan, arsip, notifikasi, dan hak akses pengguna dalam satu alur kerja yang lebih terkontrol. Pengujian *black box* memperlihatkan bahwa fungsi utama berjalan sesuai skenario kebutuhan. Kontribusi penelitian ini terletak pada integrasi proses perencanaan hingga pelaporan, bukan sekadar digitalisasi pencatatan aset.

Kata Kunci: Sistem Informasi; Sarana Prasarana; SIPRAKAR; *Iterative Incremental*; *Black box testing*

Abstract—This study develops SIPRAKAR, a web-based information system designed to support facilities and infrastructure management at the Asset and Internal Affairs Management Institute of Universitas Pamulang. The problem addressed in this study emerges from fragmented work processes, especially in work program recording, budget planning, task assignment, progress monitoring, and report preparation. This condition reduces data traceability and slows operational decision-making. The study applies the *Iterative Incremental* method through requirement analysis, design, staged implementation, testing, and module refinement. The system is developed using *Laravel*, *React*, *Inertia.js*, and *MySQL*. The development results indicate that SIPRAKAR can connect work programs, budget plans, job data, progress checklists, reports, archives, notifications, and user access control into a more manageable workflow. *Black box testing* shows that the main functions operate according to the defined requirement scenarios. The contribution of this study lies in integrating planning, budgeting, implementation, and reporting processes rather than merely converting asset records into digital form.

Keywords: Information System; Sarana Prasarana; SIPRAKAR; *Iterative Incremental*; *Black box testing*

1. PENDAHULUAN

Pengelolaan sarana dan prasarana di perguruan tinggi menuntut data yang tertata, mudah dilacak, dan dapat dipakai untuk membaca kondisi operasional secara cepat. Ketika pencatatan masih bergantung pada dokumen terpisah, hubungan antara rencana kerja, kebutuhan anggaran, pelaksanaan pekerjaan, dan laporan akhir menjadi sulit diikuti. Masalah ini tidak hanya menyangkut kerapian administrasi. Dalam praktik manajemen kampus, data yang tersebar dapat memperpanjang waktu koordinasi, membuat status pekerjaan kabur, dan melemahkan dasar pengambilan keputusan. Temuan (Annisa et al. 2023) menunjukkan bahwa sistem inventaris berbasis *web* membantu sekolah mengurangi hambatan pencatatan manual, sedangkan (Faizal & Irfan, 2025) menekankan pentingnya sistem aset kampus yang dapat menyajikan data secara lebih rapi untuk kebutuhan pengawasan.

Lembaga Pengelola Aset dan Urusan Dalam Universitas Pamulang, yang dalam penelitian ini disebut LPAUD, mengelola rangkaian pekerjaan yang cukup luas. Unit ini tidak hanya mencatat aset, tetapi juga menangani program kerja, penyusunan Rancangan Anggaran Biaya, penugasan petugas, pemantauan progres, arsip pekerjaan, dan pelaporan kegiatan. Hasil observasi pada kegiatan kerja praktik menunjukkan bahwa proses tersebut masih berjalan melalui berkas dan



komunikasi yang belum terhubung dalam satu *platform*. Program kerja dapat dibuat dalam dokumen tertentu, RAB disusun pada berkas lain, pekerjaan disampaikan secara langsung, lalu progres dicatat oleh petugas melalui catatan yang tidak selalu berada pada sumber data yang sama. Pola seperti ini membuat pimpinan membutuhkan waktu lebih lama ketika ingin mengetahui hubungan antara rencana awal, anggaran yang diajukan, pekerjaan yang berjalan, dan laporan akhir.

Penelitian tentang sistem manajemen aset berbasis *web* sudah banyak dilakukan. (Nasrul et al., 2024) membahas sistem aset untuk kebutuhan perhitungan penyusutan fiskal. (Zai et al., 2025) mengembangkan sistem manajemen aset universitas yang menekankan pendataan dan pelaporan. (Dinata et al., 2024) juga menempatkan data aset *real-time* sebagai kebutuhan penting dalam manajemen aset pendidikan. Rangkaian studi tersebut memberi dasar bahwa digitalisasi aset perlu dilakukan, tetapi kasus LPAUD memperlihatkan kebutuhan yang lebih spesifik. Akar masalahnya bukan hanya ketiadaan data aset, melainkan terputusnya alur kerja dari program kerja menuju RAB, pekerjaan, *checklist progres*, dan laporan.

Berdasarkan celah tersebut, penelitian ini mengembangkan SIPRAKAR sebagai sistem informasi pengelolaan sarana dan prasarana berbasis *web*. Sistem ini diarahkan untuk membangun keterhubungan proses, bukan sekadar menyediakan tempat penyimpanan data. Program kerja menjadi titik awal. RAB disusun berdasarkan program yang telah dibuat. RAB yang disetujui dapat diturunkan menjadi data pekerjaan. Petugas memperbarui *checklist*, lalu sistem menyajikan progres dan laporan. Dengan desain seperti itu, setiap proses memiliki jejak digital yang dapat dibaca kembali oleh admin, petugas, lembaga, dan pimpinan sesuai kewenangannya.

Pemilihan *Laravel*, *React*, *Inertia.js*, dan *MySQL* mengikuti karakter sistem yang membutuhkan pengolahan data terstruktur, antarmuka interaktif, serta kontrol hak akses. (Putra et al., 2025) menjelaskan bahwa kombinasi *Laravel* dan *React* mendukung pengembangan *web* modern karena *backend* dan *frontend* dapat dikelola dengan pembagian tanggung jawab yang jelas. Di sisi pengembangan, metode *Iterative Incremental* dipilih karena kebutuhan SIPRAKAR tidak dapat diselesaikan secara sekali jadi. (Permana et al., 2022) dan (Kusyono et al., 2024) menunjukkan bahwa pendekatan bertahap memberi ruang bagi evaluasi dan perbaikan pada tiap modul. Bagi SIPRAKAR, pola ini penting karena modul program kerja, RAB, pekerjaan, *checklist*, dan laporan saling memengaruhi. Kesalahan kecil pada satu modul dapat mengganggu alur berikutnya.

Tujuan penelitian ini adalah merancang dan membangun SIPRAKAR untuk mengintegrasikan pengelolaan program kerja, RAB, pekerjaan, *checklist progres*, laporan, arsip, notifikasi, serta manajemen pengguna pada LPAUD Universitas Pamulang. Artikel ini juga menganalisis bagaimana pendekatan *Iterative Incremental* membantu penyusunan modul secara bertahap dan bagaimana pengujian *black box* digunakan untuk memeriksa kesesuaian fungsi sistem. Dengan arah tersebut, artikel ini menawarkan pembacaan yang lebih analitis terhadap pengembangan sistem, terutama pada aspek keterlacakan proses, pengendalian hak akses, dan nilai praktis sistem bagi tata kelola sarana prasarana kampus.

2. METODE PENELITIAN

2.1 Analisis Kebutuhan

Penelitian ini menggunakan pendekatan pengembangan sistem informasi. Fokus kajian berada pada proses merancang, membangun, dan mengevaluasi aplikasi yang dibutuhkan oleh pengguna, bukan pada pengujian hubungan statistik antarvariabel. Objek penelitian adalah proses pengelolaan sarana dan prasarana pada LPAUD Universitas Pamulang. Data kebutuhan sistem diperoleh dari observasi terhadap alur kerja, wawancara dengan pihak yang memahami proses operasional, telaah dokumen kerja, dan studi pustaka. Observasi dipakai untuk melihat bagaimana program kerja, RAB, pekerjaan, *checklist progres*, dan laporan dikelola sebelum sistem dibuat. Wawancara memberi gambaran tentang kebutuhan peran pengguna, batas kewenangan, dan titik masalah yang sering muncul. Dokumentasi dipakai untuk memahami bentuk data yang selama ini digunakan, terutama format program kerja, RAB, dan laporan pekerjaan.

Kebutuhan yang ditemukan kemudian dipetakan ke dalam modul autentikasi, program kerja, RAB, pekerjaan, *checklist progres*, laporan, arsip, notifikasi, serta manajemen *role* dan *permission*. Pemetaan ini dilakukan agar setiap fitur yang dibangun tidak hanya memenuhi kebutuhan teknis, tetapi juga mengikuti alur kerja yang berjalan di LPAUD Universitas Pamulang.



2.2 Metode Pengembangan Sistem

Pengembangan sistem memakai metode *Iterative Incremental*. Pendekatan ini membagi pekerjaan ke dalam beberapa bagian kecil yang dapat diuji dan disempurnakan secara bertahap. Dalam konteks SIPRAKAR, metode ini lebih sesuai daripada pola linier karena kebutuhan pengguna dapat berubah setelah mereka melihat bentuk awal modul. (Retzlaff et al., 2025) menjelaskan bahwa praktik *iteratif* dan *incremental* membantu tim beradaptasi terhadap perubahan kebutuhan dan memasukkan umpan balik secara lebih terarah. Pada penelitian ini, prinsip tersebut diterapkan melalui pembangunan modul secara berurutan, mulai dari fondasi autentikasi sampai pelaporan dan manajemen sistem.

Tabel 1. Skema Iterasi Pengembangan SIPRAKAR

Iterasi	Fokus	Luaran yang Dicapai
1	Kebutuhan dan autentikasi	Kebutuhan pengguna dipetakan, <i>login</i> disiapkan, struktur <i>role</i> awal dibuat, dan <i>dashboard</i> dasar dibentuk.
2	Program kerja	Data rencana kegiatan, prioritas, lokasi, estimasi biaya, dan status program dapat dikelola melalui sistem.
3	RAB	Detail <i>item</i> biaya, total anggaran, pengajuan, revisi, persetujuan, dan penolakan RAB dibuat dalam alur digital.
4	Pekerjaan dan <i>checklist</i>	Data pekerjaan, petugas, <i>checklist</i> , progres otomatis, kendala, solusi, dan dokumentasi foto disiapkan.
5	Laporan dan manajemen sistem	Laporan, <i>export data</i> , arsip, notifikasi, <i>role permission</i> , <i>master data</i> , dan <i>activity log</i> dilengkapi.
6	Pengujian	Skenario black box dijalankan, kesalahan fungsi diperbaiki, dan antarmuka akhir dirapikan.

2.3 Perancangan

Tahap perancangan disusun untuk menerjemahkan kebutuhan pengguna ke dalam rancangan proses, rancangan data, dan rancangan antarmuka. Pemodelan visual digunakan agar hubungan antara aktor, fitur, dan struktur data dapat dibaca sebelum sistem diimplementasikan. Pada tahap ini, *sequence diagram* dipakai untuk melihat urutan interaksi pada proses inti, sedangkan *Entity Relationship Diagram* digunakan untuk menata hubungan data program kerja, RAB, pekerjaan, pengguna, petugas, dan *checklist*.

Perancangan SIPRAKAR menempatkan program kerja sebagai awal siklus. Dari data tersebut, sistem mengarahkan proses menuju penyusunan RAB, konversi menjadi pekerjaan, pembaruan *checklist*, hingga penyusunan laporan. Pola ini dipilih karena masalah utama pada proses manual bukan hanya pencatatan yang lambat, melainkan keterputusan data dari satu tahap ke tahap berikutnya.

2.4 Implementasi

Teknologi yang digunakan terdiri atas *Laravel* sebagai *backend*, *React* dengan *Inertia.js* sebagai *frontend*, dan *MySQL* sebagai basis data. *Laravel* dipilih karena memiliki pola *Model View Controller*, *middleware*, validasi, autentikasi, *input* dan pengelolaan *route* yang mendukung aplikasi berbasis *role*. *React* digunakan karena antarmuka SIPRAKAR membutuhkan respons cepat saat pengguna mengelola status, *filter* data, formulir RAB, dan *checklist* pekerjaan. *Inertia.js* dipakai untuk menjembatani *Laravel* dan *React* tanpa membangun *API* terpisah secara penuh. *MySQL* digunakan karena struktur data SIPRAKAR membutuhkan relasi antar tabel yang jelas.

Implementasi dilakukan secara bertahap mengikuti hasil iterasi. Modul dasar seperti *login*, *dashboard*, *role*, dan *permission* dibangun lebih awal karena menjadi fondasi kontrol akses. Setelah itu, pengembangan diarahkan pada modul program kerja, RAB, pekerjaan, *checklist progres*, laporan, arsip, dan notifikasi. Cara ini memberi ruang untuk memeriksa hubungan antarmodul sebelum seluruh sistem digunakan secara utuh.

2.5 Pengujian



Pengujian sistem menggunakan *black box testing*. Pengujian ini memeriksa fungsi dari sudut pandang pengguna melalui masukan dan keluaran yang dihasilkan sistem. Cara ini relevan karena target penelitian berada pada kelayakan fungsi, bukan pembacaan struktur kode. (Yulherniwati et al., 2025 dan Azizah et al., 2024) menempatkan *black box testing* sebagai pendekatan yang dapat menguji kesesuaian fitur dengan skenario penggunaan. Pada SIPRAKAR, skenario uji mencakup autentikasi, program kerja, RAB, pekerjaan, *checklist progres*, notifikasi, hak akses, laporan, dan arsip.

2.6 Pemeliharaan

Tahap pemeliharaan ditempatkan sebagai tindak lanjut setelah sistem diuji dan digunakan. Pada tahap ini, pengembang perlu memeriksa konsistensi data, memperbaiki kesalahan yang muncul saat sistem dipakai, menyesuaikan fitur dengan kebutuhan lembaga, dan menjaga agar hak akses tetap sesuai dengan pembagian tugas pengguna. Pemeliharaan menjadi penting karena SIPRAKAR berhubungan langsung dengan aktivitas operasional yang dapat berubah mengikuti kebijakan LPAUD.

3. HASIL DAN PEMBAHASAN

3.1 Implementasi Sistem

Hasil analisis awal memperlihatkan bahwa proses pengelolaan sarana dan prasarana di LPAUD masih memiliki fragmentasi data. Fragmentasi ini tampak pada cara program kerja, RAB, pekerjaan, dan laporan dikelola pada ruang pencatatan yang berbeda. Dampaknya terasa pada dua lapis proses. Lapis pertama berkaitan dengan pekerjaan teknis, seperti pencarian data, pembaruan status, dan rekap laporan. Lapis kedua berkaitan dengan keputusan manajerial, karena pimpinan tidak selalu dapat membaca status pekerjaan berdasarkan data yang sama dengan admin atau petugas. Kondisi tersebut menunjukkan bahwa masalah utama bukan hanya lambatnya pencatatan, tetapi lemahnya hubungan antardata.

Pada proses RAB, kelemahan paling menonjol berada pada status dokumen. Ketika pengajuan, revisi, persetujuan, dan penolakan tidak tercatat dalam sistem terpusat, pengguna harus menelusuri riwayat komunikasi atau dokumen terdahulu. Cara ini membuka ruang salah tafsir. RAB yang sebenarnya masih perlu diperbaiki dapat dianggap telah siap. RAB yang sudah disetujui dapat sulit ditemukan kembali. Dalam konteks pengelolaan sarana prasarana, status anggaran tidak dapat dipisahkan dari status pekerjaan. Tanpa hubungan tersebut, pekerjaan lapangan berisiko berjalan tanpa dasar data yang lengkap.

SIPRAKAR dirancang untuk menjawab masalah tersebut melalui alur kerja yang berurutan. Program kerja ditempatkan sebagai sumber awal data. Dari program kerja, admin dapat menyusun kebutuhan biaya dalam RAB. RAB yang masuk ke tahap *review* akan memiliki status yang dapat dibaca oleh pihak berwenang. Ketika RAB disetujui, sistem dapat mengubah program kerja menjadi pekerjaan yang siap dijalankan. Setelah pekerjaan dibuat, petugas dapat memperbarui *checklist progres*. Data *checklist* inilah yang menjadi dasar perhitungan progres dan laporan. Rancangan ini membuat setiap tahap tidak berdiri sendiri.

Integrasi tersebut menjadi pembeda SIPRAKAR dari sistem inventaris yang hanya menekankan pencatatan barang. Studi (Nasrul et al., 2024 dan Zai et al., 2025) memberi gambaran bahwa sistem aset berbasis *web* dapat mendukung pendataan dan pelaporan. SIPRAKAR mengambil ruang yang lebih operasional, yakni menghubungkan rencana kerja dan pelaksanaan pekerjaan. Dengan desain seperti itu, data tidak berhenti sebagai daftar aset, tetapi bergerak menjadi dasar koordinasi. Admin dapat membaca rencana, lembaga dapat memantau pengajuan, petugas dapat melihat tugas, dan pimpinan dapat meninjau laporan.

Dari sisi hak akses, sistem memakai pendekatan *role* dan *permission*. Pendekatan ini penting karena tidak semua pengguna memiliki kepentingan yang sama terhadap data. Superadmin mengelola pengaturan utama, *role*, *permission*, dan *master data*. Admin menangani program kerja, RAB, pekerjaan, laporan, dan arsip. Lembaga atau pimpinan berfokus pada peninjauan dan persetujuan. Staff lebih banyak berinteraksi dengan pekerjaan dan *checklist progres*. Pembagian tersebut memperkecil kemungkinan akses yang tidak sesuai. (Effendri & Ma'sum, 2025) menekankan bahwa pengelolaan aset digital perlu memperhatikan struktur aplikasi dan kebutuhan

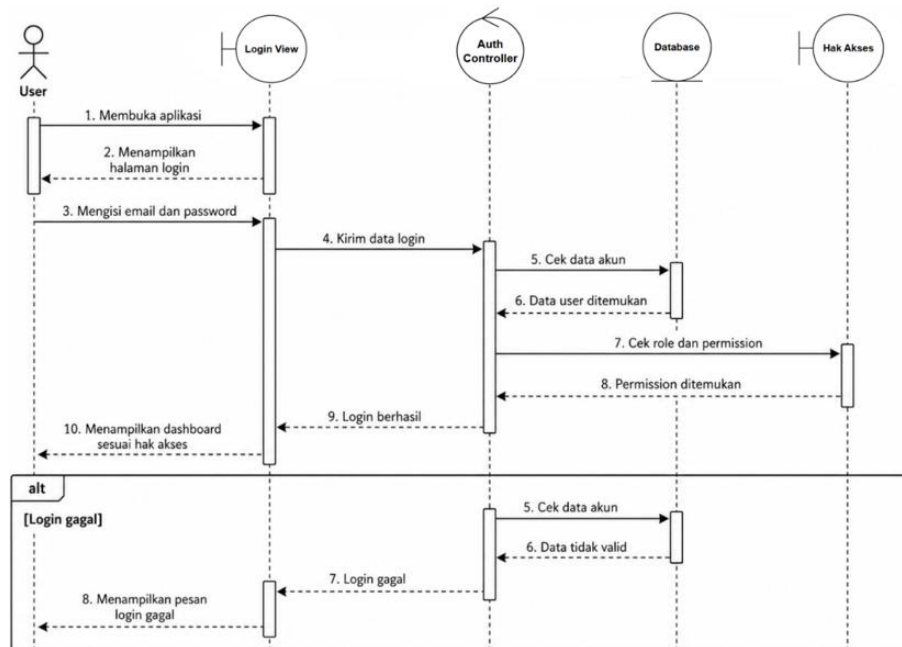
organisasi, karena aplikasi yang baik harus mengikuti cara kerja lembaga, bukan memaksa lembaga mengikuti pola teknis yang kaku.

3.1.1 Hasil Perancangan Sistem

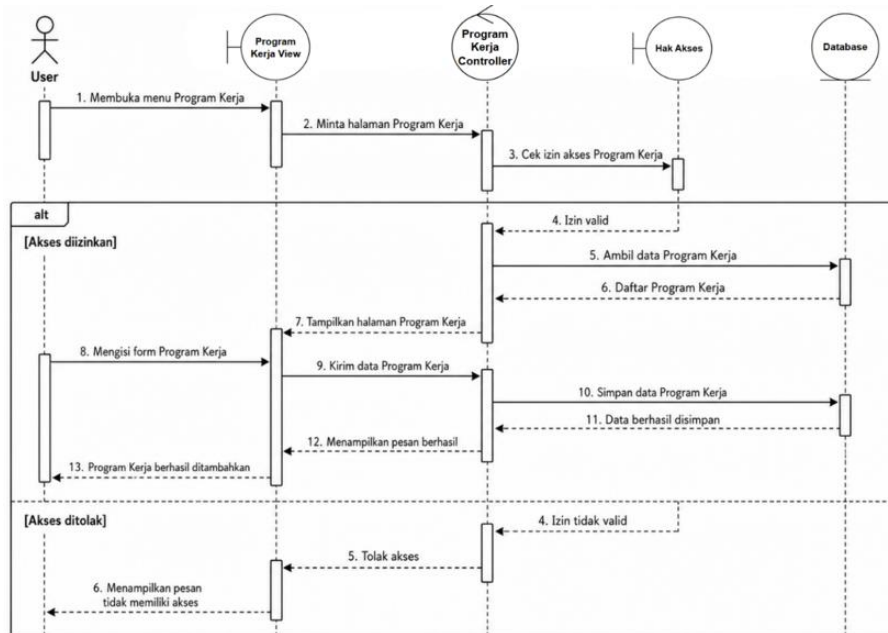
a. Sequence diagram

Perancangan SIPRAKAR diarahkan pada keteraturan alur komunikasi antara pengguna, tampilan sistem, *controller*, hak akses, dan basis data. *Sequence diagram* digunakan untuk memperlihatkan bagaimana proses utama berjalan dari sisi interaksi sistem. Diagram ini dipilih karena alur SIPRAKAR banyak melibatkan validasi data, pemeriksaan *role*, dan perubahan status pada beberapa modul yang saling berhubungan.

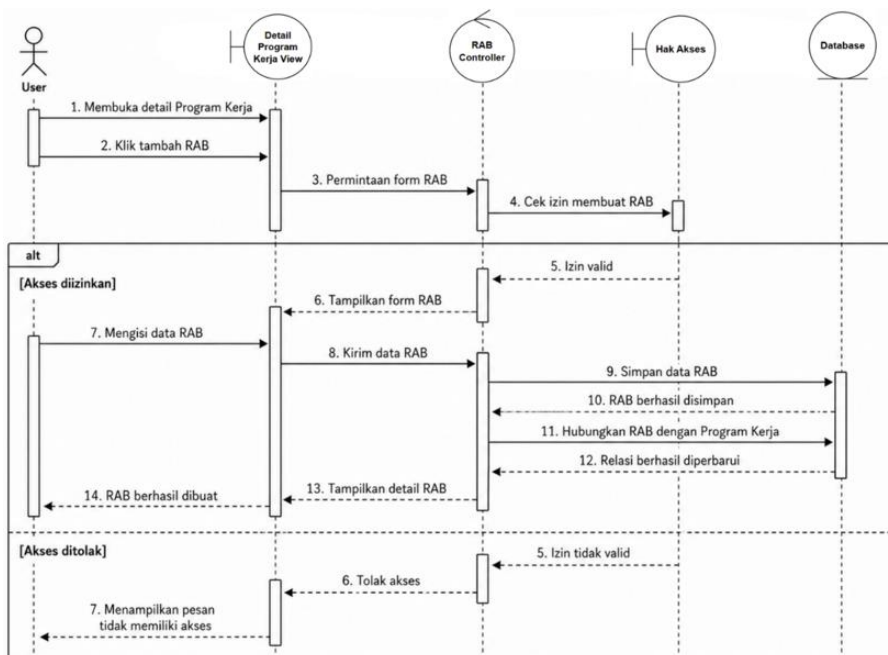
Sequence diagram yang ditampilkan mencakup alur *login*, pengelolaan program kerja, pembuatan RAB, konversi program kerja menjadi pekerjaan, serta pembaruan progres pekerjaan. Kelima proses tersebut dipilih karena menjadi rangkaian inti dalam SIPRAKAR. Jika salah satu proses tidak berjalan dengan baik, hubungan antara perencanaan, anggaran, pelaksanaan, dan pelaporan akan ikut terganggu.



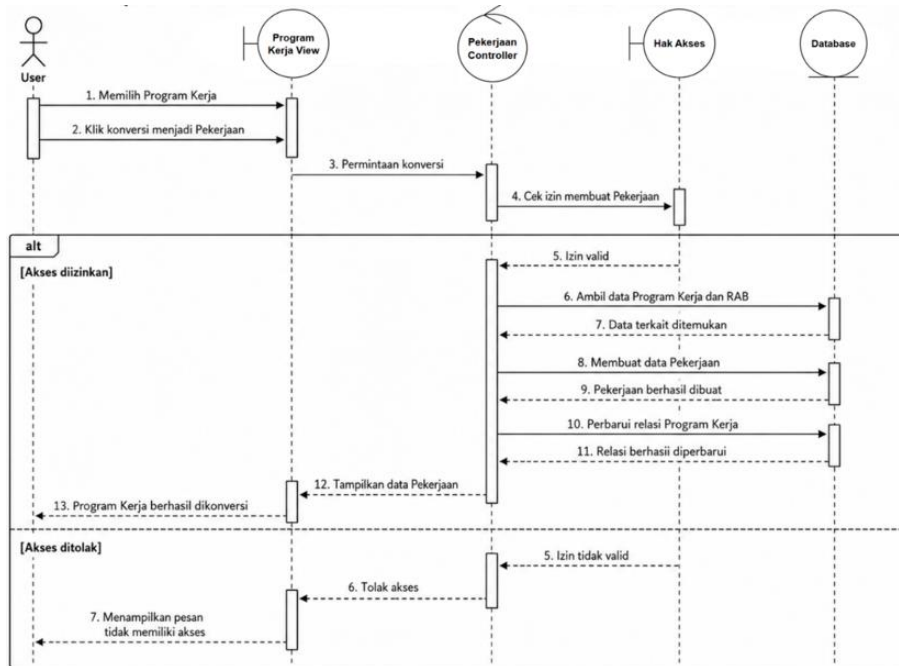
Gambar 1. Sequence diagram Login dan Hak Akses SIPRAKAR



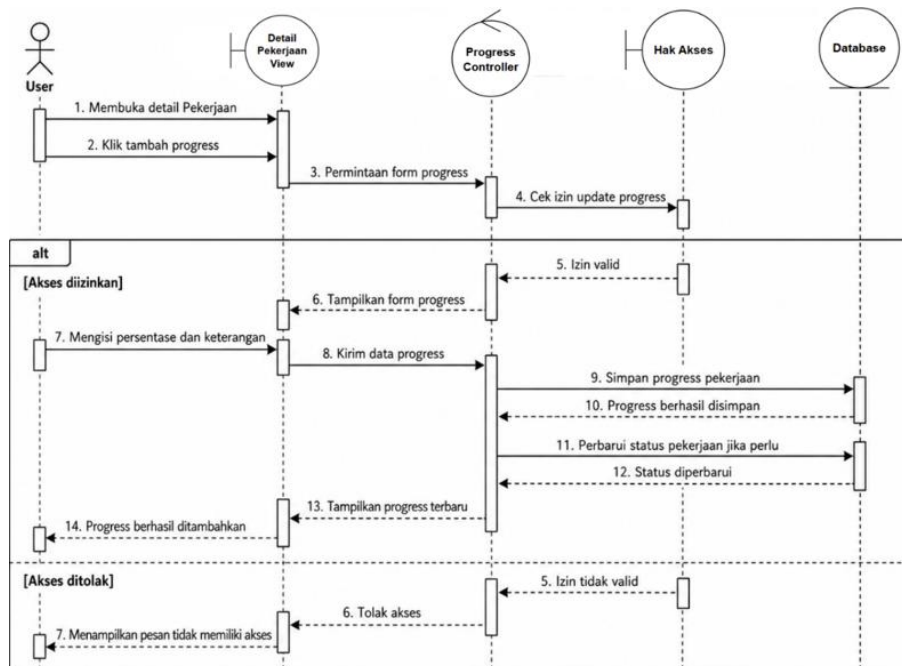
Gambar 2. *Sequence diagram* Pengelolaan Program Kerja



Gambar 3. *Sequence diagram* Pembuatan RAB



Gambar 4. Sequence diagram Konversi Program Kerja Menjadi Pekerjaan



Gambar 5. Sequence diagram Pembaruan Progres Pekerjaan

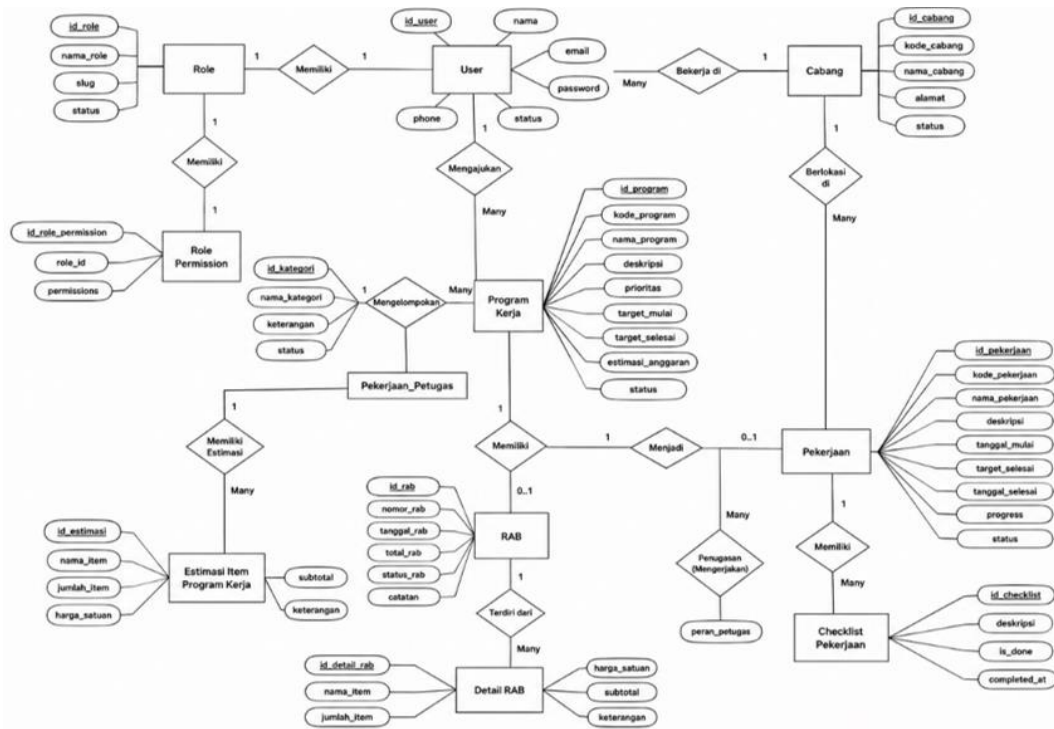
Pada proses *login*, sistem memeriksa data akun terlebih dahulu melalui basis data. Ketika akun valid, sistem melanjutkan pemeriksaan *role* dan *permission* sebelum pengguna diarahkan ke *dashboard*. Alur ini menunjukkan bahwa autentikasi dalam SIPRAKAR tidak hanya memastikan identitas pengguna, tetapi juga menentukan batas menu yang boleh diakses.

Pada proses program kerja, pembuatan RAB, konversi pekerjaan, dan pembaruan progres, sistem selalu menempatkan pemeriksaan izin sebelum perubahan data dilakukan.

Pola tersebut membantu menjaga agar aktivitas penting hanya dijalankan oleh pengguna yang berwenang. Bagi LPAUD, rancangan ini cukup penting karena data program kerja, RAB, dan progres pekerjaan berkaitan langsung dengan tanggung jawab operasional.

b. Entity Relationship Diagram

Perancangan basis data SIPRAKAR disusun untuk menjaga keterhubungan data dari tahap perencanaan sampai pelaporan. ERD digunakan untuk memperlihatkan entitas utama, atribut penting, dan relasi yang membentuk struktur sistem. Diagram ini tidak hanya berfungsi sebagai rancangan teknis, tetapi juga menunjukkan bagaimana alur kerja LPAUD diterjemahkan ke dalam model data yang lebih teratur.



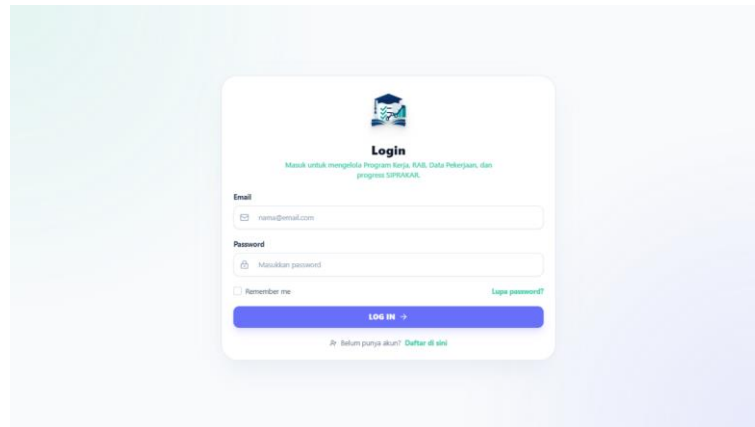
Gambar 6. Entity Relationship Diagram SIPRAKAR

ERD SIPRAKAR memuat beberapa entitas utama, yaitu *User*, *Role*, *Role permission*, *Cabang*, *Program Kerja*, *Kategori*, *Estimasi Item Program Kerja*, *RAB*, *Detail RAB*, *Pekerjaan*, *Pekerjaan Petugas*, dan *Checklist Pekerjaan*. *Program Kerja* menjadi dasar awal proses, *RAB* menghubungkan kebutuhan anggaran, *Pekerjaan* mencatat pelaksanaan, sedangkan *Checklist Pekerjaan* menyimpan perkembangan tugas. Relasi ini membuat setiap data dapat ditelusuri dari rencana awal sampai hasil pekerjaan.

3.1.2 Tampilan Antarmuka Sistem

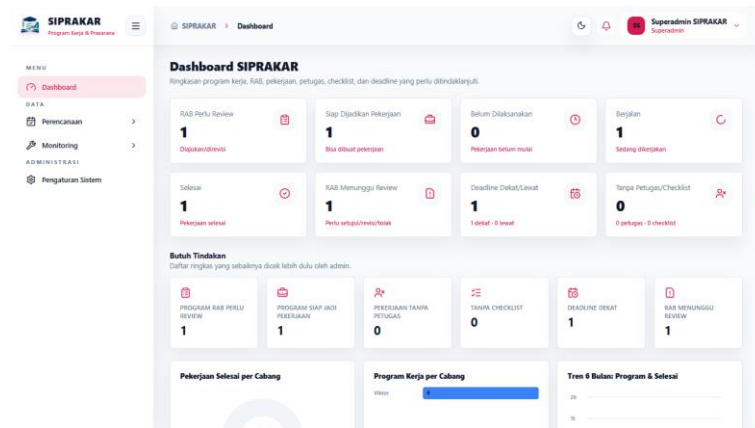
Tampilan antarmuka SIPRAKAR dibuat untuk memudahkan pengguna membaca status pekerjaan dan mengakses fitur sesuai kewenangannya. Bagian ini menampilkan beberapa halaman utama yang mewakili alur sistem, mulai dari *login*, *dashboard*, *program kerja*, *RAB*, *data pekerjaan*, *detail pekerjaan*, sampai pengaturan sistem. Setiap tampilan tidak hanya berfungsi sebagai halaman *input*, tetapi juga menjadi titik kontrol terhadap status dan relasi data.

a. Tampilan Login SIPRAKAR



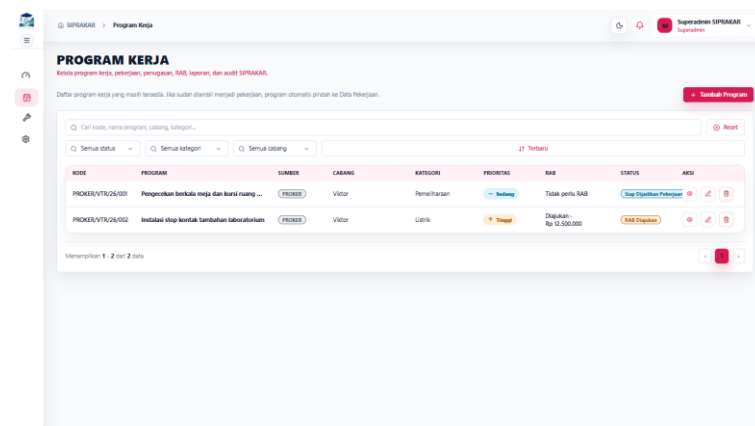
Gambar 7. Tampilan *Login* SIPRAKAR

b. Tampilan *Dashboard* SIPRAKAR



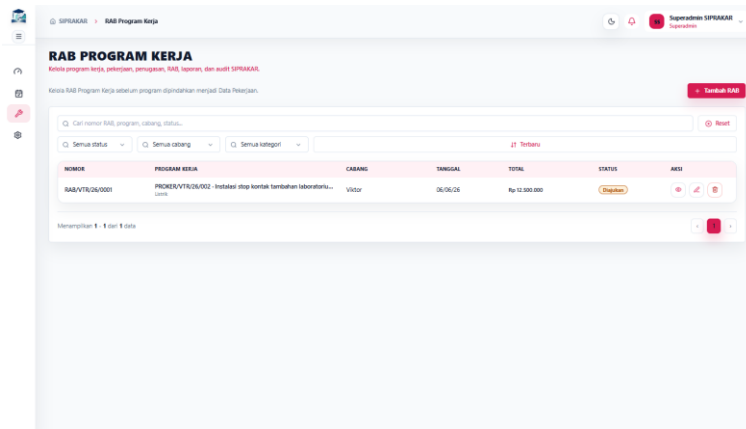
Gambar 8. Tampilan *Dashboard* SIPRAKAR

c. Tampilan Menu Program Kerja



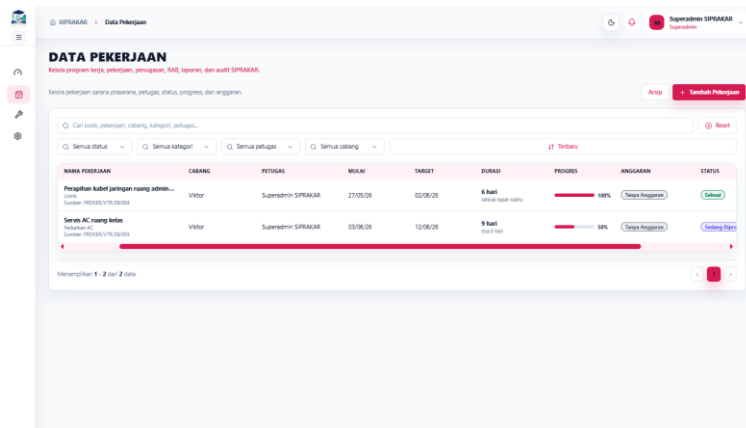
Gambar 9. Tampilan Menu Program Kerja

d. Tampilan Menu RAB Program Kerja



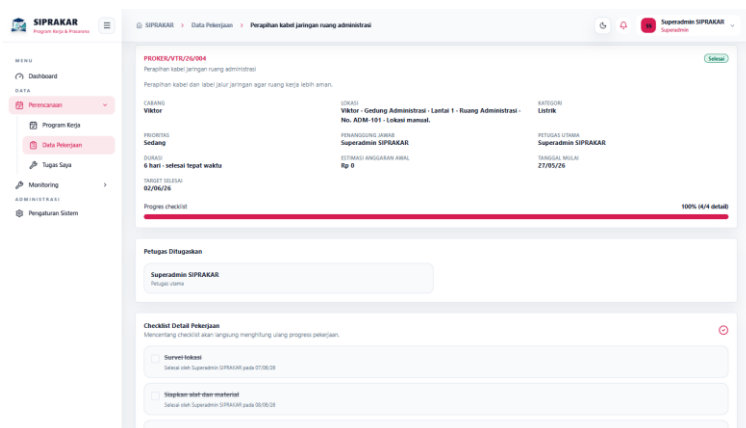
Gambar 10. Tampilan Menu RAB Program Kerja

e. Tampilan Menu Data Pekerjaan



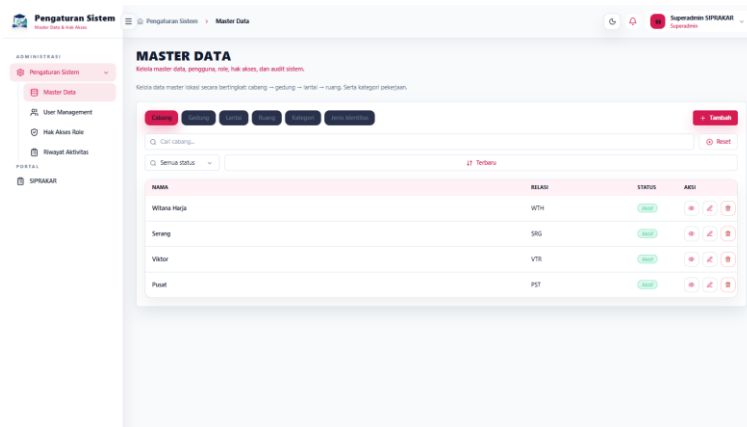
Gambar 11. Tampilan Menu Data Pekerjaan

f. Tampilan Detail Pekerjaan dan Checklist progres



Gambar 12. Tampilan Detail Pekerjaan dan Checklist progres

g. Tampilan Menu Pengaturan Sistem



Gambar 13. Tampilan Menu Pengaturan Sistem

Modul program kerja berfungsi sebagai ruang awal perencanaan. Data yang disimpan tidak hanya nama kegiatan, tetapi juga prioritas, lokasi, estimasi biaya, status, dan keterangan pendukung. Modul ini memberi gambaran yang lebih terarah kepada admin ketika menyusun rencana. Pada proses manual, estimasi biaya dapat terpisah dari uraian program. Di SIPRAKAR, estimasi tersebut melekat pada data program sehingga hubungan antara rencana dan kebutuhan biaya lebih mudah dibaca. Cara ini juga membantu ketika program kerja harus dikonversi menjadi pekerjaan setelah tahapan anggaran selesai.

Modul RAB memberi struktur pada proses anggaran. Setiap *item* dapat dicatat dengan jumlah, satuan, harga, dan total biaya. Sistem menempatkan RAB dalam alur pengajuan dan *review* sehingga status dokumen tidak hanya diketahui melalui komunikasi informal. Pada tahap ini, nilai sistem tidak terletak pada kemampuan menghitung total biaya saja. Nilai yang lebih penting adalah keterlacakan keputusan. Pengguna dapat melihat apakah RAB masih menunggu *review*, perlu revisi, disetujui, atau ditolak. Keterlacakan semacam ini membantu lembaga menghindari kebingungan saat beberapa program berjalan pada waktu yang berdekatan.

Modul pekerjaan dan *checklist progres* berperan sebagai jembatan antara dokumen perencanaan dan pelaksanaan lapangan. Pekerjaan yang sudah terbentuk dapat ditugaskan kepada petugas. Petugas kemudian memperbarui *checklist* sesuai perkembangan tugas. Sistem menghitung progres berdasarkan *checklist* yang telah diselesaikan. Pendekatan ini membuat pemantauan lebih objektif karena progres tidak hanya bergantung pada laporan lisan. Admin dapat membaca bagian mana yang sudah selesai, bagian mana yang tertunda, dan kendala apa yang tercatat pada pekerjaan tertentu.

Modul laporan, arsip, notifikasi, dan *activity log* melengkapi kebutuhan pengawasan. Laporan memungkinkan pengguna menyaring data berdasarkan status, waktu, atau kategori tertentu. Arsip membantu menyimpan pekerjaan yang tidak lagi aktif tanpa menghilangkan jejaknya. Notifikasi memberi peringatan ketika ada perubahan status atau penugasan baru. *Activity log* menyimpan riwayat aktivitas pengguna sehingga tindakan penting dapat ditelusuri kembali. Dalam tata kelola sarana prasarana, fitur seperti ini penting karena pekerjaan lapangan sering melibatkan lebih dari satu pihak dan membutuhkan catatan yang dapat dipertanggungjawabkan.

3.2 Pengujian Sistem

Pengujian sistem dilaksanakan dengan metode *black box testing* untuk memastikan seluruh fitur utama berjalan sesuai skenario. Pengujian ini dilakukan dari sudut pandang pengguna sehingga fokusnya berada pada kesesuaian masukan, proses, dan keluaran yang muncul pada antarmuka sistem. Dengan cara ini, sistem dapat diperiksa berdasarkan fungsi yang benar-benar digunakan dalam kegiatan LPAUD.

Tabel 2. Hasil Pengujian Black Box SIPRAKAR

Modul	Arah Pengujian	Indikator Keberhasilan	Temuan
Autentikasi	<i>Login valid, password salah, akun tidak aktif, dan akses dashboard.</i>	Pengguna valid masuk sistem, kesalahan ditolak, dan <i>dashboard</i> tampil sesuai peran.	Sesuai skenario.
Program kerja	Tambah, ubah, hapus, <i>filter</i> , status, prioritas, lokasi, dan estimasi biaya.	Data tersimpan, dapat dicari, diperbarui, dan ditampilkan kembali tanpa kehilangan relasi.	Sesuai skenario.
RAB	Pembuatan <i>item</i> , total biaya, pengajuan, <i>review</i> , revisi, persetujuan, dan penolakan.	Status berubah mengikuti alur dan keputusan <i>reviewer</i> tercatat dalam sistem.	Sesuai skenario.
Pekerjaan	Konversi program, penugasan petugas, status pekerjaan, dan detail lokasi.	Pekerjaan terbentuk dari program yang sesuai dan petugas dapat melihat tugasnya.	Sesuai skenario.
<i>Checklist</i>	Tambah <i>checklist</i> , perbarui kondisi, catatan kendala, solusi, dan dokumentasi.	Progres dihitung otomatis dan status selesai muncul ketika <i>checklist</i> terpenuhi.	Sesuai skenario.
Laporan dan arsip	<i>Filter</i> data, rekap laporan, <i>export CSV</i> , arsip pekerjaan, dan pelacakan kembali.	Data dapat disaring, direkap, diekspor, dan disimpan sebagai arsip.	Sesuai skenario.
Hak akses	Pembatasan menu dan fungsi berdasarkan <i>role</i> dan <i>permission</i> .	Pengguna hanya melihat menu yang sesuai dengan kewenangannya.	Sesuai skenario.

Hasil pengujian memperlihatkan bahwa fungsi utama SIPRAKAR berjalan sesuai kebutuhan yang telah dirumuskan. Pada modul autentikasi, sistem dapat membedakan pengguna valid, kesalahan *password*, dan akun tidak aktif. Pada modul program kerja dan RAB, perubahan status dapat mengikuti alur yang ditetapkan. Pada modul pekerjaan dan *checklist*, sistem mampu memperbarui progres secara otomatis berdasarkan *checklist* yang selesai. Hasil ini menunjukkan bahwa rancangan integrasi data sudah berjalan pada tingkat fungsional.

Dari sudut pandang kualitas perangkat lunak, hasil *black box testing* memberi dasar awal bahwa SIPRAKAR dapat digunakan pada lingkungan kerja LPAUD. Azizah et al. (2024) menunjukkan bahwa *black box testing* dapat menemukan bagian fungsi yang tidak berjalan sesuai skenario pengguna. Dalam penelitian ini, pengujian tidak diarahkan untuk menilai efisiensi kode, keamanan mendalam, atau beban akses tinggi. Ruang uji masih berada pada perilaku fitur. Karena itu, hasil berhasil pada pengujian fungsional perlu dibaca secara proporsional. Sistem layak secara fungsi utama, tetapi tahap pengembangan berikutnya tetap perlu menguji keamanan, performa, dan pengalaman pengguna secara lebih rinci.

Secara kritis, kelebihan SIPRAKAR berada pada kemampuan menyusun jejak kerja dari perencanaan hingga laporan. Sistem seperti ini dapat membantu lembaga mengurangi ketergantungan pada komunikasi informal. Meski demikian, keberhasilan implementasi tidak dapat hanya dibebankan pada aplikasi. LPAUD perlu menyiapkan aturan *input* data, standar penamaan program, jadwal pembaruan progres, dan pembagian tanggung jawab antar pengguna. Tanpa kebiasaan *input* yang konsisten, sistem terintegrasi tetap dapat menghasilkan data yang kurang akurat. Teknologi memberi struktur, sementara disiplin organisasi menentukan kualitas data yang masuk.

Implikasi praktis penelitian ini terlihat pada perubahan cara kerja. Admin tidak perlu menelusuri banyak dokumen untuk memahami status program kerja. Petugas dapat membaca tugas dan memperbarui *checklist* melalui sistem. Pimpinan dapat melihat rekap kegiatan dengan lebih



cepat. Lembaga juga memperoleh catatan status RAB yang lebih transparan. Manfaat ini sejalan dengan temuan (Annisa et al., 2023; Faizal & Irfan, 2025; Dinata et al., 2024) yang menekankan pentingnya sistem berbasis *web* untuk memperbaiki akses data dan kualitas pelaporan.

Implikasi akademik penelitian ini berada pada perluasan fokus kajian sistem aset. Banyak penelitian sebelumnya menempatkan aset sebagai objek pendataan. SIPRAKAR memperlihatkan bahwa pengelolaan sarana prasarana kampus juga membutuhkan sistem kerja yang menghubungkan rencana, anggaran, tugas, progres, dan laporan. Dengan kata lain, kontribusi sistem tidak hanya diukur dari banyaknya data yang tersimpan, tetapi dari seberapa jelas hubungan antarproses dapat ditelusuri. Arah ini dapat menjadi dasar bagi penelitian berikutnya yang ingin menggabungkan manajemen aset, manajemen pekerjaan, dan evaluasi layanan kampus.

Keterbatasan penelitian perlu ditempatkan secara jelas. Pengujian yang dilakukan masih berfokus pada *black box testing*. Sistem belum diuji melalui beban akses besar, audit keamanan, uji kepuasan pengguna, atau pengukuran efisiensi waktu sebelum dan sesudah implementasi. Penelitian berikutnya dapat menambahkan *usability testing*, *user acceptance testing*, pengujian performa, dan integrasi dengan sistem keuangan universitas. Pengembangan *mobile friendly* juga penting karena petugas lapangan sering bekerja jauh dari komputer. Dengan pengembangan lanjutan, SIPRAKAR dapat menjadi sistem operasional yang lebih matang dan lebih mudah diterapkan pada unit pengelolaan sarana prasarana lain.

4. KESIMPULAN

Penelitian ini menghasilkan SIPRAKAR sebagai sistem informasi berbasis *web* untuk mendukung pengelolaan sarana dan prasarana pada LPAUD Universitas Pamulang. Sistem dikembangkan dengan metode *Iterative Incremental* sehingga setiap modul dapat dibangun melalui tahapan kecil, diperiksa, lalu disempurnakan berdasarkan kebutuhan pengguna. SIPRAKAR menghubungkan program kerja, RAB, pekerjaan, *checklist progres*, laporan, arsip, notifikasi, dan manajemen pengguna dalam satu alur kerja yang lebih mudah ditelusuri. Dengan rancangan tersebut, proses pengelolaan sarana prasarana tidak lagi hanya bergantung pada dokumen terpisah dan komunikasi manual.

Hasil implementasi memperlihatkan bahwa *Laravel*, *React*, *Inertia.js*, dan *MySQL* dapat digunakan untuk membangun sistem yang memiliki kontrol hak akses, alur persetujuan RAB, pembaruan progres, dan pelaporan. Pengujian *black box* menunjukkan bahwa fitur utama berjalan sesuai skenario fungsional yang telah ditentukan. Kontribusi utama penelitian ini terletak pada integrasi proses, karena SIPRAKAR menempatkan perencanaan, penganggaran, pelaksanaan, pemantauan, dan pelaporan sebagai satu rangkaian data. Pengembangan berikutnya disarankan mencakup pengujian keamanan, performa, *usability*, *user acceptance testing*, integrasi keuangan, dan optimasi akses melalui perangkat bergerak.

REFERENCES

- Annisa, R., Rahayuningsih, P. A., & Anna, A. (2023). Perancangan sistem informasi inventaris sarana dan prasarana sekolah berbasis *web*. *Infotek: Jurnal Informatika dan Teknologi*, 6(1), 60-70. <https://doi.org/10.29408/jit.v6i1.7356>
- Azizah, D. N., Mahendar, I. A., Alfatih, M. F., Anwar, S. I., Al Hapid, N. M., Wicaksono, A., & Mindara, G. P. (2024). Analysis and testing of the Combox *web* application system using *black box testing* with the equivalence partitioning method. *International Journal of Electrical Engineering, Mathematics and Computer Science*, 1(4), 37-43. <https://doi.org/10.62951/ijeemcs.v1i4.118>
- Dinata, R. K., Hasdyna, N., & Hijriana, N. (2024). *Web*-based asset management information system for higher education. *Journal of Engineering and Technology*, 6(1), 41-48. <https://jurnal.uniki.ac.id/index.php/jet/article/view/419>
- Effendri, A., & Ma'sum, H. (2025). Perancangan aplikasi aset manajemen menggunakan framework *Laravel* di PT Dirgantara Indonesia (IAe). *Jurnal Informatika dan Teknik Elektro Terapan*, 13(2), 1-9. <https://doi.org/10.23960/jitet.v13i2.6405>
- Faizal, L., & Irfan, I. (2025). Perancangan sistem informasi manajemen aset kampus berbasis *web* menggunakan metode waterfall. *Jurnal Ilmiah Sistem Informasi dan Teknik Informatika*, 8(2), 208-216. <https://doi.org/10.57093/jisti.v8i2.342>



JRIIN : Jurnal Riset Informatika dan Inovasi
Volume 4, No. 3 Tahun 2026
ISSN 3025-0919 (media online)
Hal 797-810

- Kusyono, A. Z. A., Adi, T. N., & Thohiroh, E. L. (2024). Pengembangan *website* edukasi kesehatan balita dengan menggunakan metode *Iterative Incremental*. *KLIK: Kajian Ilmiah Informatika dan Komputer*, 5(1), 263-274. <https://doi.org/10.30865/klik.v5i1.1962>
- Nasrul, N., Saptono, H., Wibowo, E., & Amalia, A. (2024). Rancang bangun sistem informasi manajemen aset berbasis *web* untuk menghitung penyusutan fiskal. *Jurnal Informatika Terpadu*, 10(1), 66-72. <https://doi.org/10.54914/jit.v10i1.1086>
- Permana, K. D., Fauzi, R., & Suakanto, S. (2022). Pengembangan *backend* investasi berbasis *website* pada ekosistem digital Ihya dengan metode *Iterative Incremental*. *JURIKOM: Jurnal Riset Komputer*, 9(5), 1226-1233. <https://doi.org/10.30865/jurikom.v9i5.4830>
- Putra, F. P. E., Efendi, R. W., Tamam, A. B., & Pramadi, W. A. (2025). Trends and best practices in *API*-based *web* development using *Laravel* and *React*. *Brilliance: Research of Artificial Intelligence*, 5(1), 223-233. <https://doi.org/10.47709/brilliance.v5i1.5971>
- Retzlaff, N., Spohrer, K., Fagerholm, F., & Mendez, D. (2025). Measuring agility in software development teams. *Frontiers in Computer Science*, 7, 1626456. <https://doi.org/10.3389/fcomp.2025.1626456>
- Yulherniwati, Kasmar, A. F., Hadi, R., Humaira, H., & Asri, E. (2025). Black-box testing pada sistem evaluasi capaian pembelajaran berdasarkan outcome based education. *JITSI: Jurnal Ilmiah Teknologi Sistem Informasi*, 6(2), 203-212. <https://doi.org/10.62527/jitsi.6.2.279>
- Zai, Y. Y. M., Nanda, A. P., & Riyantori, R. (2025). Perancangan sistem manajemen aset berbasis *web* Universitas Mercubaktijaya. *Jurnal Pustaka Data*, 5(2), 438-445. <https://doi.org/10.55382/jurnalpustakadata.v5i2.1487>