

Automasi Dokumentasi Aset Melalui Mekanisme *Dynamic Report Generation* dengan Transformasi Visual secara *Real-Time*

Akbar Madya¹

¹ Departemen Sistem Informasi, Fakultas Sains dan Teknologi, Universitas Labuhanbatu

INFORMASI ARTIKEL

Diterima 3 Februari 2026
Direvisi 13 Februari 2026
Diterbitkan 19 Februari 2026

Kata kunci:

MySarPras,
SQLite;
iText7;
ApachePOI;
Automasi Laporan

ABSTRAK

Manajemen sarana dan prasarana sering mengalami kesulitan menyinkronkan administrasi data dengan dokumentasi visual. Penelitian ini mengembangkan aplikasi MySarPras berbasis Android dengan arsitektur SQLite dan integrasi library iText7 serta Apache POI. Kebaruan penelitian ini terletak pada mekanisme alur transformasi visual yang mengoptimalkan manajemen memori melalui teknik *downsampling* untuk menghasilkan laporan PDF/Excel secara proporsional. Hasil pengujian *BlackBox* menunjukkan seluruh fungsi CRUD berjalan stabil dengan tingkat interoperabilitas tinggi melalui fitur *Implicit Intent*. Hasil pengujian performa menunjukkan sistem mampu menghasilkan laporan secara instan dengan durasi 1,8 detik untuk 10 data hingga 9,3 detik untuk 100 data aset dengan ukuran file mencapai 30,3 MB. MySarPras berhasil mendigitalisasi proses pendataan menjadi lebih akuntabel dan efisien tanpa ketergantungan internet. Pemanfaatan teknologi *mobile* ini menawarkan sarana alternatif potensial untuk menjawab kebutuhan birokrasi manajemen aset di berbagai organisasi secara efisien.

Asset Documentation Automation Through *Dynamic Report Generation Mechanism* with Visual Transformation in *Real-Time*

ARTICLE INFO

Received February 3, 2026
Revised February 13, 2026
Published February 19, 2026

Keyword:

MySarPras;
SQLite;
iText7;
Apache POI;
Report Automation

ABSTRACT

Facilities and infrastructure management often have difficulty synchronizing data administration with visual documentation. This research develops an Android-based MySarPras application with SQLite architecture and integration of iText7 library and Apache POI. The novelty of this research lies in the visual transformation flow mechanism that optimizes memory management through downsampling techniques to produce PDF/Excel reports proportionally. The results of the Black Box test show that all CRUD functions run stably with a high level of interoperability through the *Implicit Intent* feature. The results of the performance test showed that the system was able to generate reports instantly with a duration of 1.8 seconds for 10 data to 9.3 seconds for 100 data assets with a file size of up to 30.3 MB. MySarPras has succeeded in digitizing the data collection process to be more accountable and efficient without internet dependence. The use of mobile technology offers a potential alternative means to efficiently address the bureaucratic needs of asset management in various organizations.

This work is licensed under a [Creative Commons Attribution 4.0](https://creativecommons.org/licenses/by/4.0/)



Corresponding Author:

Corresponding Akbar Madya, Universitas Labuhanbatu
Email: akbarmadya89@gmail.com

1. PENDAHULUAN

Manajemen aset sarana dan prasarana adalah bagian penting dari keberlangsungan sebuah organisasi [1]. Pada dasarnya, manajemen aset adalah mengelola aset untuk menghasilkan manfaat yang paling besar dengan biaya yang paling rendah serta menjaga agar aset tidak punah, kecuali jika memang sebaiknya untuk dihancurkan maupun dihapuskan [2]. Namun, pengalaman di lapangan menunjukkan bahwa metode inventarisasi konvensional masih menghadapi masalah besar karena adanya perbedaan informasi antara kondisi fisik aset yang sebenarnya dan catatan administratif. Ketika pelaporan dilakukan secara manual, masalah utama muncul. Data yang tersedia ataupun nama dan kode barang seringkali tidak didukung oleh bukti visual yang akurat, sehingga membuat proses verifikasi dan audit menjadi lebih sulit [3].

Karena laporan pendataan lapangan saat ini bergantung pada pemindahan data manual dari lapangan ke perangkat komputer, proses penyusunan laporan birokrasi masih sangat lama. Selain itu, mengelola dokumentasi visual dalam skala besar pada perangkat dengan sumber daya terbatas sering menyebabkan inkonsistensi tata letak dan penurunan kinerja aplikasi [4].

Untuk mengatasi permasalahan tersebut, penelitian ini mengusulkan sebuah mekanisme automasi dokumentasi aset melalui mekanisme *dynamic report generation* dengan transformasi visual secara *real-time* [5]. Berbeda dengan sistem inventarisasi statis, penelitian ini memfokuskan pada aspek pengambilan informasi (*information retrieval*) dari database relasional lokal SQLite yang memanfaatkan fitur CRUD (*Create, Read, Update, Delete*) kemudian diintegrasikan langsung dengan *engine* pelaporan iText7 dan Apache POI melalui aplikasi Android Studio.

CRUD adalah suatu pengkodean yang dapat membuat, membaca, memperbarui serta menghapus data [6]. SQLite merupakan basis data *open source* yang tertanam dalam android dan memiliki sintaks SQL, fungsi transaksi, dan fungsi *prepared statement*. Selain itu, saat dijalankan, hanya membutuhkan sedikit memori [7]. Menurut Lowgie (2025), arsitektur iText dimaksudkan untuk menjadi perpustakaan pemrosesan PDF yang fleksibel dengan pembagian tugas yang jelas melalui integrasi komponen *kernel*, *layout*, dan *renderer* [8]. Sedangkan Apache POI merupakan suatu pengembangan sistem untuk membaca, memodifikasi serta menconvert data menjadi data dalam bentuk berkas (.xlsx) [9]. Penelitian sebelumnya telah melakukan pengembangan proyek manajemen aset serupa, namun fungsionalitas terkait luaran (*output*) dokumen otomatis pada pengerjaan tersebut belum tersedia. Oleh karena itu, penelitian ini melakukan peningkatan (*upgrade*) melalui mekanisme *dynamic report generation* untuk memaksimalkan efisiensi aplikasi yang dirancang.

Salah satu kebaruan (*novelty*) dalam penelitian ini adalah mekanisme alur integrasi data yang menyinkronkan akuisisi citra gambar melalui *FileProvider* dengan manajemen memori gambar menggunakan (*downsampling*), sehingga transformasi visual ke dalam dokumen formal dapat dilakukan secara otomatis dan stabil pada perangkat seluler [10]. Sistem ini melindungi data dalam lingkungan seluler dengan pola *Data Access Object* (DAO) dan integrasi *FileProvider* [11]. Dengan digitalisasi alur kerja, penelitian ini dapat membuat dokumen siap cetak secara instan dari perangkat android, meningkatkan efisiensi sistem informasi infrastruktur. Justru itu, teknologi *mobile* ini menawarkan sebagai sarana alternatif yang sangat berpotensi untuk menjawab kebutuhan tersebut [12].

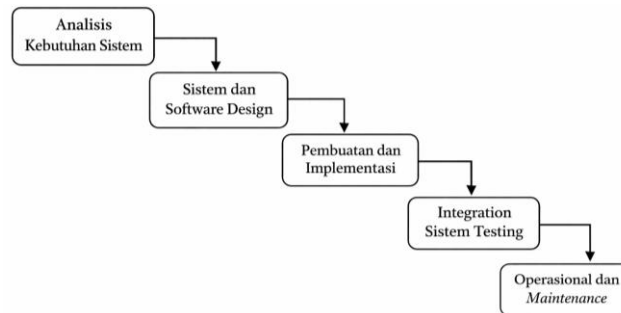
2. METODE

Penelitian ini menggunakan pendekatan rekayasa perangkat lunak berbasis penelitian terapan (*applied research*) dengan tujuan mengusulkan dan mengimplementasikan sebuah mekanisme pelaporan aset berbasis visual pada lingkungan *mobile* [13]. Metodologi yang digunakan dirancang tidak hanya untuk membangun sistem, tetapi juga untuk mengevaluasi mekanisme yang diusulkan secara fungsional dan visual.

2.1. Tahapan Pengembangan Sistem

Sebagai model klasik dalam rekayasa perangkat lunak, metode pengembangan *waterfall* dipilih untuk penelitian ini karena memiliki tahapan yang terstruktur dan sistematis dan cocok untuk pengembangan aplikasi dengan ruang lingkup yang sudah jelas [14]. Seperti yang ditunjukkan pada

Gambar 1, model *waterfall* terdiri dari beberapa tahapan berurutan, yaitu: analisis kebutuhan, desain sistem, implementasi, pengujian, dan pemeliharaan [15][16].



Gambar 1. *Waterfall Diagram*

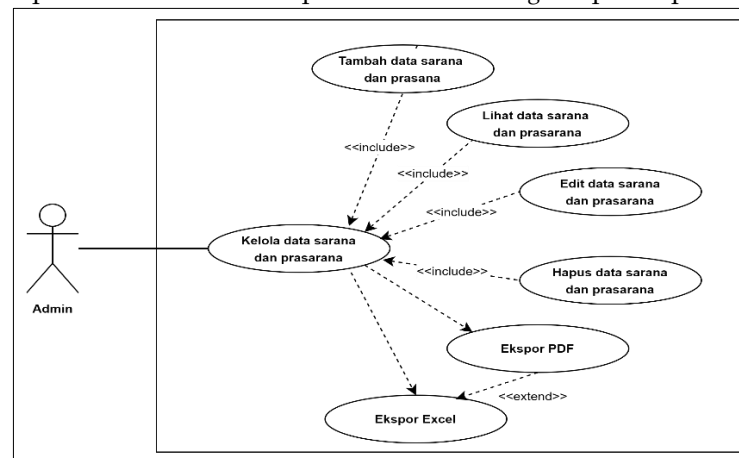
- a. **Analisis Kebutuhan Sistem**
Penelitian dimulai dengan menemukan masalah informasi yang menghambat proses audit antara catatan administratif dan kondisi fisik aset di lapangan. Tahap ini menetapkan persyaratan fungsional seperti pengumpulan data melalui kamera, penyimpanan SQLite, dan otomasi laporan PDF/Excel. Selain itu, persyaratan tidak fungsional termasuk standar keamanan *FileProvider* dan optimasi memori citra dengan menggunakan *library Glide*.
- b. **Sistem dan Software Design**
Pada tahap desain, sistem arsitektur dibangun menggunakan pola *Data Access Object (DAO)* untuk membedakan logika database SQLite dari antarmuka pengguna. Selain mesin logika pelaporan *iText7* dan *Apache POI*, perancangan mencakup alur kerja pengecekan izin kamera (*permission check*), dan desain antarmuka pengguna yang diarahkan untuk mengambil posisi foto landscape untuk memastikan tata letak dokumen tetap menarik.
- c. **Pembuatan dan Implementasi**
Untuk mengintegrasikan fitur kamera ke *FileProvider* sesuai dengan protokol keamanan API Android terbaru, implementasi dilakukan menggunakan *Java Native*. Membangun mekanisme penskalaan otomatis gambar ke dalam tabel laporan tanpa merusak aspek rasio serta menjamin bahwa tekstual data dengan referensi lokasi (*path*) gambar dalam database adalah tujuan utama pengkodean.
- d. **Integration Sistem Testing**
Pengujian menyeluruh dilakukan untuk memverifikasi stabilitas alur kerja dari input hingga ekspor serta sinkronisasi data visual secara *real-time*. Dalam uji interoperabilitas, memfokuskan pada fitur *Auto-Open* menggunakan *Implicit Intent* digunakan untuk memastikan bahwa dokumen laporan dapat dideteksi dan dibuka secara instan oleh aplikasi pihak ketiga seperti *PDF Reader* atau spreadsheet pada perangkat pengguna.
- e. **Operasional dan Maintenance**
Pada tahap akhir, aplikasi diimplementasikan pada lingkungan organisasi untuk meningkatkan produktivitas aset manajemen dengan menyediakan dokumen laporan siap cetak langsung dari perangkat seluler. Aplikasi ini juga dilakukan pemeliharaan secara berkala untuk memastikan bahwa aplikasi kompatibel dengan sistem pembaruan operasi android dan melacak kinerja sistem dalam menangani dokumentasi *visual aset* yang sangat besar.

Secara keseluruhan, tahapan penelitian *waterfall* membantu proses pengembangan aplikasi secara berurutan dan terstruktur, mulai dari analisis kebutuhan hingga evaluasi akhir. Ini memungkinkan hasil akhir penelitian untuk sepenuhnya memenuhi tujuan penelitian secara efektif.

3. HASIL DAN PEMBAHASAN

Hasil implementasi dan analisis menyeluruh aplikasi *MySarPras*, yang dikembangkan menggunakan *Android Studio* berbasis *Java* dan menggunakan *SQLite* sebagai database persistence lokal, disajikan di bagian ini. Berdasarkan uji coba sistem, proses *CRUD (Create, Read, Update, Delete)*

terbukti berjalan dengan baik. Pola DAO melindungi hubungan antara data teks dan rute gambar aset pada penyimpanan internal perangkat. Hasil pengujian menunjukkan bahwa mesin pelaporan, yang menggunakan iText7 dan Apache POI, dapat membuat laporan otomatis dengan tingkat tata letak yang sangat tepat. Gambar 2. menampilkan *Use Case Diagram* pada aplikasi MySarPras.



Gambar 2. *Use Case Diagram* aplikasi mobile MySarPras

3.1. Implementasi Kerangka Aplikasi

Aplikasi MySarPras dibuat dengan sukses menggunakan Android Studio dan bahasa pemrograman Java. Aplikasi ini menggunakan database SQLite sebagai media penyimpanan data lokal yang efektif. Aplikasi ini dimaksudkan untuk mendigitalisasi data manajemen aset seperti nama barang, jenis, lokasi, dan kondisi. Ini juga memiliki fitur unggulan seperti generasi laporan otomatis dan dokumentasi visual (foto). Struktur proyek di Android Studio disusun secara modular. Beberapa komponen utama meliputi:

- Activity*: *MainActivity*, *AddSaranaActivity*, dan *PreviewLaporanActivity*.
- Model*: Kelas *Sarana* yang merepresentasikan entitas data dan *path* gambar.
- Database Helper*: Kelas *DatabaseHelper* yang menangani operasi CRUD (*Create*, *Read*, *Update*, *Delete*) serta penyimpanan URI gambar.
- Reporting Engine*: Mekanisme alur integrasi data library iText7 (PDF) dan Apache POI (Excel) untuk penyusunan laporan secara otomatis.
- Image Handler*: Penggunaan *Glide* untuk manajemen memori saat melakukan transformasi visual citra aset.

3.2. Implementasi Fitur CRUD dan Automasi Laporan

Melalui sinkronisasi informasi administratif dan dokumentasi aset visual secara *real-time*, fokus utama dalam pengembangan aplikasi ini adalah menjamin integritas data. Untuk meningkatkan skalabilitas dan keamanan sistem, implementasi arsitektur perangkat lunak yang melibatkan penggunaan kelas *DatabaseHelper* sebagai lapisan abstraksi yang membedakan logika data dasar dari antarmuka pengguna.

- Create* (Menambah data & akuisi visual)

Mekanisme ini tidak hanya menangani persistensi data tekstual, tetapi juga secara terstruktur mengintegrasikan referensi rute lokasi gambar yang diperoleh dari pembelian kamera. Dengan menggunakan objek *ContentValues*, proses pemetaan data juga dikenal sebagai *data mapping* yang dilakukan dengan tepat ke dalam kolom tabel SQLite. Ini memastikan bahwa integritas antara informasi administratif dan bukti visual aset tetap terjaga secara konsisten. Implementasi kode ditujukan pada potongan kode:

```
public void addSarana(String nama, String lokasi, String kondisi, String photoPath) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put("nama_barang", nama);
    values.put("lokasi", lokasi);
    values.put("kondisi", kondisi);
    values.put("path_foto", photoPath); // Menyimpan referensi file gambar
    db.insert("sarana", null, values);
    db.close();
}
```

b. *Read* (Manajemen memori citra/gambar)

Dalam proses menampilkan data (*Read*), masalah teknis utama terletak pada pengelolaan memori saat memuat banyak aset visual ke dalam komponen *RecyclerView*. Aplikasi ini mengintegrasikan perpustakaan *Glide*, yang mampu melakukan *downsampling* dan *caching* secara otomatis, memastikan bahwa kinerja perangkat tetap stabil dan efisien, meskipun harus memberikan daftar sarana dan dokumentasi gambar. Penerapan ini ditujukan pada potongan kode:

```
// Contoh pengambilan data menggunakan Cursor
Cursor cursor = db.rawQuery("SELECT * FROM sarana ORDER BY id_sarana DESC", null);
```

c. *Update* dan *Delete* (Integritas data)

Untuk memastikan identitas objek yang diubah dalam database benar, proses pembaruan data menggunakan *id_sarana* yang unik. Saat proses penghapusan data dimulai, sistem secara otomatis menghapus semua referensi yang terkait dengan database, yang mencegah redundansi data dan menjaga integritas ruang penyimpanan internal perangkat. Kode pada fitur ini ditujukan pada potongan kode:

```
// Update data berdasarkan ID
db.update("sarana", values, "id_sarana = ?", new String[]{String.valueOf(id)});
// Delete data
db.delete("sarana", "id_sarana = ?", new String[]{String.valueOf(id)});
```

d. Automasi Laporan (*Dynamic Report Generation*)

Fitur unggulan ini bekerja dengan mengekstraksi seluruh dataset dari SQLite dan kemudian disusun secara sistematis ke dalam tabel dokumen menggunakan mesin pelaporan *iText7*. Proses ini tidak hanya mengandalkan fungsi bawaan *library*, melainkan menerapkan mekanisme integrasi alur kerja yang mengelola *path* citra dari *database* lokal, memprosesnya melalui *FileProvider*, dan menerapkan logika transformasi visual. Mekanisme ini memastikan bahwa foto aset terintegrasi ke dalam sel tabel secara otomatis dengan ukuran yang proporsional dan stabil, sehingga menjaga integritas aspek rasio pada dokumen formal meskipun dijalankan pada perangkat dengan spesifikasi terbatas. Berikut potongan kode:

```
// Logika penyisipan gambar ke PDF
ImageData data = ImageDataFactory.create(img_path);
Image img = new Image(data);
img.setAutoScale(true); // Transformasi visual otomatis sesuai sel tabel
table.addCell(img);
```

Untuk memvalidasi efisiensi mekanisme *dynamic report generation*, dilakukan pengujian performa sistem dalam menghasilkan laporan PDF dan Excel menggunakan instrumentasi pengujian internal.

Pengujian dilakukan dengan memvariasikan jumlah entri data aset yang menyertakan lampiran gambar. Hasil pengukuran durasi proses ekspor dapat dilihat pada Tabel 1 dan 2.

Tabel 1. Hasil pengujian kecepatan ekspor laporan PDF

Jumlah Data Aset	Ukuran File	Durasi Proses (detik)	Hasil Uji
10 Data	2.5 MB	1.8 s	Berhasil
20 Data	5.1 MB	2.5 s	Berhasil
50 Data	18.2 MB	5.4 s	Berhasil
100 Data	30.3 MB	9.3 s	Berhasil

Tabel 2. Hasil pengujian kecepatan ekspor laporan Excel

Jumlah Data Aset	Ukuran File	Durasi Proses (detik)	Hasil Uji
10 Data	321 KB	1.0 s	Berhasil
20 Data	583 KB	1.5 s	Berhasil
50 Data	987 KB	3.1 s	Berhasil
100 Data	1.8 MB	7.8 s	Berhasil

Menurut hasil pengujian pada Tabel 1 dan Tabel 2, proses pembuatan laporan Excel relatif lebih singkat dengan rata-rata 7,8 detik untuk 100 data. Jika dibandingkan dengan laporan PDF relatif lebih lama dengan durasi 9,3 detik. Hasil ini menunjukkan bahwa peningkatan jumlah data berbanding lurus dengan durasi proses, namun tetap berada dalam batas toleransi stabilitas perangkat seluler. Hal ini dikarenakan mekanisme penulisan data Apache POI mengintegrasikan objek ke dalam skema XML laporan secara langsung, sementara iText7 membutuhkan proses kalkulasi layout halaman yang lebih kompleks untuk menjaga tampilan visual yang tepat.

Sistem penyimpanan data aplikasi MySarPras dioperasikan melalui basis data relasional SQLite. Tabel 3 menampilkan skema tabel yang dimaksudkan untuk mendukung operasional secara luring (*offline*) dan integritas data inventaris.

Tabel 3. Struktur tabel *database* aplikasi MySarPras

Nama Field	Tipe Data	Keterangan
id_sarana	Integer	Primary Key (Auto Increment)
nama_barang	String	Nama aset sarana yang didata
jenis_barang	String	Kategori atau jenis aset
lokasi	String	Lokasi penyimpanan fisik aset
kondisi	String	Status kondisi fisik (baik/rusak)
path_photo	String	Alamat penyimpanan file gambar di memori perangkat

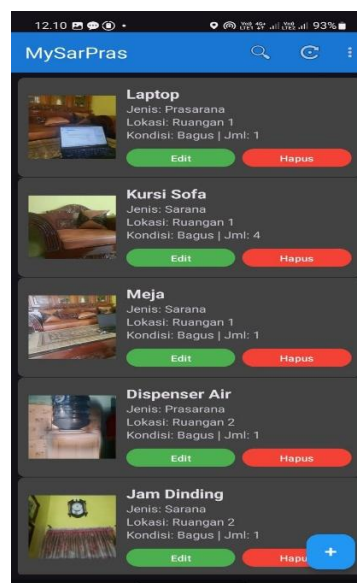
Hasil pengujian fungsional menunjukkan bahwa penggunaan SQLite sangat efektif untuk penyimpanan data lokal secara *offline*. Namun, ditemukan catatan penting tentang elemen keamanan data jangka panjang. Karena SQLite menyimpan file *database* secara langsung di dalam direktori internal perangkat, itu sangat bergantung pada kondisi fisik perangkat. Semua data inventaris yang telah dimasukkan tidak dapat dipulihkan kembali jika perangkat mengalami kerusakan sistem (*brick*) atau hilang. Untuk itu dibutuhkan pencadangan otomatis pada perangkat seperti *drive* eksternal. Hal ini menegaskan bahwa, meskipun sistem ini unggul dalam kemudahan pengoperasian tanpa internet, terdapat konsekuensi teknis terkait dengan elemen redundansi data yang menjadi batasan penelitian ini.

3.3. Implementasi UI (User Interface)

Dalam aplikasi MySarPras, implementasi antarmuka pengguna (UI) dibangun menggunakan pendekatan materi desain yang sederhana, mudah dipahami, dan mudah digunakan oleh pengguna dari berbagai latar belakang. Setiap elemen antarmuka pengguna dibangun menggunakan komponen bawaan Android Studio seperti *TextView*, *EditText*, *Button*, *RecyclerView*, *ImageView*, dan *FloatingActionButton*. Untuk memastikan konsistensi antarmuka dan mudah dibaca, pengguna dapat memahami fungsi aplikasi dengan cepat tanpa instruksi khusus.

a. Halaman Utama (Main Activity)

Halaman utama menampilkan daftar sarana data dalam bentuk daftar vertikal yang digunakan *RecyclerView*. Setiap item menampilkan nama, jenis, lokasi, dan kondisi barang, dan juga memiliki thumbnail foto aset sebagai dokumentasi visual. Setiap item juga memiliki tombol edit dan hapus. Di pojok kanan bawah, terdapat tombol *FloatingActionButton* (FAB) dengan ikon tambah (+). Tombol ini dapat digunakan untuk membuka halaman *AddSaranaActivity*, halaman tambah data sarana baru. Gambar 3 menampilkan *dashboard* aplikasi MySarPras.



Gambar 3. Tampilan halaman utama (*dashboard*)

b. Form Tambah Data MySarPras

Tampilan form tambah data (*activity_tambah.xml*) memungkinkan pengguna mengisi nama barang, jenis barang, lokasi, dan kondisi dengan beberapa *EditText*. Untuk mengambil sarana foto secara langsung, halaman ini memiliki fitur kamera integrasi. Di bagian bawah, ada tombol *save*, yang akan memproses data teks dan jalur file gambar ke database SQLite dengan menggunakan fungsi *insertSarana()*.

c. Form Edit Data MySarPras

Halaman edit (*activity_edit.xml*) memiliki tampilan yang mirip dengan halaman tambah, tetapi seluruh *EditText* sudah terisi otomatis dengan sarana data yang dipilih melalui *Intent Extra*. Pengguna dapat mengubah teks atau mengganti foto sarana setelah menekan tombol Simpan Perubahan untuk memulai fungsi *updateSarana()* pada database.

d. Layout Item MySarPras

item_sarana.xml adalah nama file untuk setiap item yang ditampilkan dalam daftar, file ini dirancang secara responsif dan ringkas sehingga nyaman dilihat di berbagai ukuran layar.

1. *ImageView* untuk menampilkan pratampilan (*preview*) foto sarana.
2. *TextView* untuk menampilkan detail nama barang, jenis, lokasi, dan kondisi.
3. Dua tombol aksi (ikon *Edit* dan *Delete*) untuk manajemen data secara cepat.

e. Mekanisme Ekspor dan Navigasi Laporan

Mesin pelaporan iText7 dan Apache POI dengan sistem operasi android mengintegrasikan data untuk ekspor ke format PDF dan Excel (Gambar 4 dan Gambar 5). *Intent* mengatur navigasi antar halaman, sementara *Implicit Intent* memanfaatkan mekanisme interoperabilitas untuk memfasilitasi akses eksternal ke dokumen. Setelah berkas berhasil dibuat di penyimpanan lokal, sistem dapat melakukan pemanggilan otomatis (*auto-open*) terhadap aplikasi pembaca dokumen pihak ketiga melalui mekanisme ini. Alur ini memastikan bahwa pengguna dapat memverifikasi hasil transformasi visual secara langsung tanpa harus mencari berkas di dalam direktori penyimpanan.

LAPORAN DATA SARANA & PRASARANA
 Dicetak pada: Fri Feb 13 12:01:42 GMT+07:00 2026

Foto	Nama Barang	Jenis	Lokasi	Kondisi	Jml
	Laptop	Prasarana	Ruangan 1	Bagus	1
	Kursi Sofa	Sarana	Ruangan 1	Bagus	4
	Meja	Sarana	Ruangan 1	Bagus	1
	Dispenser Air	Prasarana	Ruangan 2	Bagus	1
	Jam Dinding	Sarana	Ruangan 2	Bagus	1
	Keranjang Sampah	Prasarana	Ruangan 1	Bagus	2
	Piala	Sarana	Ruangan 1	Bagus	50
	Lemari	Prasarana	Ruangan 1	Bagus	1
	Buku	Prasarana	Ruangan 2	Bagus	20

1/2

Foto	Nama Barang	Jenis	Lokasi	Kondisi	Jml
	Pena	Sarana	Ruangan 2	Bagus	5

Gambar 4. Laporan PDF

	B	C	D	E	F	G
1	Nama Sari	Jenis	Lokasi	Kondisi	Jumlah	
2	1	Laptop	Prasarana	Ruangan1	Bagus	1
3	2	Kursi Sofa	Sarana	Ruangan1	Bagus	4
4	3	Meja	Sarana	Ruangan1	Bagus	1
5	4	Dispenser	Prasarana	Ruangan 2	Bagus	1
6	5	Jam Dindir	Sarana	Ruangan 2	Bagus	1
7	6	Keranjang	Prasarana	Ruangan 1	Bagus	2
8	7	Piala	Sarana	Ruangan 1	Bagus	50
9	8	Lemari	Prasarana	Ruangan 1	Bagus	1
10	9	Buku	Prasarana	Ruangan 2	Bagus	20
11	10	Pena	Sarana	Ruangan 2	Bagus	5
12	11	Proyektor	Prasarana	Ruangan 3	Bagus	10
13	12	Papan Tuli	Sarana	Ruangan 4	Bagus	5
14	13	Pot bunga	Prasarana	Ruangan 5	Bagus	7
15	14	Kipas Angi	Sarana	Ruangan 6	Bagus	8
16	15	AC	Sarana	Ruangan 7	Bagus	90
17	16	Lampu	Sarana	Ruangan 8	Bagus	1
18	17	Pengharur	Sarana	Ruangan 9	Bagus	2
19	18	Spidol	Prasarana	Ruangan 1	Bagus	5
20	19	Penghapus	Prasarana	Ruangan 1	Bagus	7
21	20	Kursi	Prasarana	Ruangan 1	Bagus	8

Gambar 5. Laporan Excel

3.4. *BlackBox Testing*

Metode pengujian *BlackBox*, yang fokus pada fungsionalitas sistem tanpa melihat struktur kode internal program, digunakan untuk menguji aplikasi MySarPras. Seluruh fitur, mulai dari manajemen database hingga otomasi laporan, dapat digunakan sesuai dengan kebutuhan

pengguna, itulah tujuan utama dari pengujian ini. Setiap fungsi penting diuji, ini termasuk menambahkan data ke sarana dengan menggunakan foto, menampilkan data di *RecyclerView*, mengedit data, dan menghapus data.

Selain itu, skenario pengujian juga menguji fitur validasi input, seperti mengosongkan kolom yang diperlukan. Selain itu, aspek persistensi data juga diuji untuk memastikan tampilan data tetap konsisten setelah aplikasi ditutup dan dibuka kembali. Untuk memastikan interoperabilitas fitur *auto-open* melalui *Implicit Intent*, setiap pengujian dilakukan secara langsung di kedua perangkat android fisik maupun emulator.

Tabel 4. Hasil pengujian *Blackbox testing*

Fitur	Hasil yang diharapkan	Hasil uji
Tambah data dan foto	Data dan file gambar tersimpan di SQLite & penyimpanan lokal	Sesuai
Tambah (kosong)	Muncul pesan validasi atau <i>error</i> bahwa input wajib diisi	Sesuai
Tampil data	Data sarana muncul di halaman utama lengkap dengan <i>thumbnail</i> foto	Sesuai
Edit data dan foto	Data dan foto berubah sesuai dengan pembaruan yang dilakukan	Sesuai
Hapus data	Data dan referensi file gambar hilang dari daftar dan database	Sesuai
Fitur <i>Auto-Open</i>	Laporan terbuka otomatis melalui <i>Implicit Intent</i> segera setelah ekspor selesai	Sesuai
Ekspor PDF/Excel	Sistem berhasil menggenerasi laporan dengan transformasi visual yang proporsional	Sesuai
Buka ulang aplikasi	Data tetap tersimpan dan tidak terjadi kehilangan referensi <i>path</i> gambar	Sesuai

Berdasarkan hasil pengujian yang disajikan pada Tabel 4, setiap fitur utama aplikasi MySarPras telah memenuhi kriteria keberhasilan fungsional secara menyeluruh. Penerapan logika transformasi visual dengan teknik *downsampling* terbukti efektif menghasilkan dokumen yang akuntabel tanpa distorsi gambar maupun kendala memori. Selain itu, penggunaan *Implicit Intent* berhasil meningkatkan interoperabilitas sistem, memberikan kemudahan bagi pengguna untuk mengakses laporan secara instan melalui aplikasi pihak ketiga.

4. KESIMPULAN

Penelitian ini berhasil mengimplementasikan aplikasi MySarPras yang mampu menggabungkan mekanisme penciptaan laporan dinamis berbasis android untuk meningkatkan efisiensi dokumentasi aset dengan akuntabel. Sistem dapat membuat laporan PDF dan Excel yang stabil dengan tingkat interoperabilitas tinggi melalui fitur *AutoOpen*, yang menyinkronkan data SQLite dengan mesin pelaporan iText7 dan Apache POI. Meskipun basis data SQLite menawarkan keunggulan dalam operasional secara luring (*offline*), ada keterbatasan pada penyimpanan lokal, yang berarti bahwa data dapat hilang jika perangkat fisik rusak atau hilang. Oleh karena itu, untuk menjamin keamanan dan redundansi data jangka panjang, maka pengembangan selanjutnya harus mengintegrasikan mekanisme sinkronisasi ke peladen (*cloud database/server*) ataupun fitur pencadangan otomatis.

DAFTAR PUSTAKA

- [1] Afifah Rashif and Ria Arifianti, "Efektivitas Pemeliharaan Aset Di Kompleks Sekretariat Ukm Barat Dan Timur," *J. Lentera Bisnis*, vol. 14, no. 1, pp. 233–241, 2025, doi: 10.34127/jrlab.v14i1.1330.
- [2] R. A. Aryani Soemitro and H. Suprayitno, "Pemikiran Awal tentang Konsep Dasar Manajemen Aset

- Fasilitas," *J. Manajemen Aset Infrastruktur Fasilitas*, vol. 2, no. 0, pp. 1-14, 2018, doi: 10.12962/j26151847.v2i0.4225.
- [3] E. Wenardi, A. Wija, and D. Hajar, "Implementasi Sistem ERP ODOO untuk Optimalisasi Pencatatan Transaksi pada Ani Mart," *Median. Indones. J. Creat. Bus. Technol.*, vol. 1, no. 1, pp. 40-56, 2025, doi: 10.64425/ffpwk954.
- [4] S. Cathasa, R. Rijadi, S. Suakanto, and J. Barat, "Development Of An Information System For Asset Management," *J. INOVTEK POLBENG -SERI Inform.*, vol. 9, no. 2, pp. 940-952, 2024.
- [5] D. Irawan, P. Studi, S. Informasi, I. Komputer, and U. M. Metro, "Rancang bangun sistem informasi manajemen aset berbasis web," *JMSI*, vol. 6, no. 2, pp. 133-141, 2025.
- [6] I. S. F. D. Artha; and R. W. A. Iktisom, "Peningkatan Kemampuan Coding Anak Usia Remaja Dengan Metode Crud Generator Berbasis Web Dengan Analisa Database," *J. PEDAMAS (Pengabdi. Kpd. Masy.)*, vol. 1 (2), pp. 331-337, 2023.
- [7] R. B. D. Putra, E. S. Budi, and A. R. Kadafi, "Perbandingan Antara SQLite, Room, dan RBDLiTe Dalam Pembuatan Basis Data pada Aplikasi Android," *JURIKOM (Jurnal Ris. Komputer)*, vol. 7, no. 3, p. 376, 2020, doi: 10.30865/jurikom.v7i3.2161.
- [8] D. Arito, J. Priyanga, A. Pinandito, and M. A. Akbar, "Analisis Perbandingan Performa PDF Generator Library iText, PDFBox, dan AndroidPdfDocument Pada Pengembangan Aplikasi Android (Studi Kasus : Invoice Generator)," *J. Pengemb. Teknol. Inf. dan Ilmu Komputer(JPTIIK)*, vol. 9, no. 11, pp. 1-9, 2025.
- [9] K. L. Bawankule and N. B. Raut, "Design and implementation of massive MYSQL data intelligent export system to excel by using Apache -POI libraries," *IOSR J. Comput. Eng.*, vol. 16, no. 5, pp. 58-65, 2014, doi: 10.9790/0661-16545865.
- [10] M. Z. Asiari, "Analisis Kinerja Sistem Auto Scaling Pada Sistem Web Server Berbasis Clustering Menggunakan Sistem Virtual," *Univ. Hasanuddin*, pp. 3-35, 2021.
- [11] W. Royani Syafril, "Perancangan Sistem Informasi Manajemen Aset Berbasis Web pada Divisi Teknologi Informasi PAM JAYA," *J. Teknol. Dan Sist. Inf. Bisnis Vol.*, vol. 7, no. 4, pp. 527-533, 2025, doi: <https://doi.org/10.47233/jteksis.v7i4.2261>.
- [12] D. Kurnia, "Pengembangan Aplikasi Mobile Pengelola Data Sarana Berbasis Android Studio Dengan Fitur Crud Menggunakan SQLite," *J. Gemilang Inform.*, vol. 3, no. 2, pp. 32-38, 2025, [Online]. Available: <https://www.journal.hdgi.org/index.php/>
- [13] A. Yahya, L. N. Amali, and I. R. Padiku, "Pengembangan Sistem Aplikasi Manajemen Aset Berbasis Android Di Universitas Negeri Gorontalo," *Diffus. J. Syst. Inf. Technol.*, vol. 4, no. 1, pp. 92-102, 2024.
- [14] R. S. Ferri Rizaldi, Jihan Tshivana, Kharlin Ahmad Hanafi Siregar, Novia Anggraini, "Model Waterfall dalam Pengembangan Perangkat Lunak : Tinjauan Literatur tentang Efektivitas dan Keterbatasannya," *SIGMA J. Teknol. Pelita Bangsa*, vol. 15, no. 1, 2025.
- [15] B. Fachri, C. Rizal, and Supiyandi, "Penerapan Metode Waterfall Dalam Perancangan Sistem Informasi Merdeka Belajar Kampus Merdeka Berbasis Web," *J. Komput. Teknol. Inf. dan Sist. Inf.*, vol. 2, no. 3, pp. 591-597, 2024, doi: 10.62712/juktisi.v2i3.147.
- [16] Y. S. Rahayu, Y. Saputra, and D. Irawan, "Implementasi Metode Waterfall Pada Pengembangan Sistem Informasi Mobile E-Disarpus," *Zo. J. Sist. Inf.*, vol. 6, no. 2, pp. 523-534, 2024.