

# Rancang Bangun Aplikasi Monitoring Peminjaman Aset pada Diskominfo Kabupaten Magetan

Hayu Nimalasari<sup>1</sup>, I Made Suartana<sup>2</sup>

<sup>1,2</sup>(Teknik Informatika/Teknik Informatika, Universitas Negeri Surabaya)

[124051204243@mhs.unesa.ac.id](mailto:124051204243@mhs.unesa.ac.id)

[2imadesuartana@unesa.ac.id](mailto:2imadesuartana@unesa.ac.id)

**Abstrak**— Dinas Komunikasi dan Informatika (Diskominfo) Kabupaten Magetan mengelola aset TIK untuk mendukung e-Government, namun prosedur peminjaman antarunit kerja/OPD masih konvensional. Kondisi ini menyebabkan birokrasi persetujuan yang lambat 2–3 hari, tumpang tindih jadwal, dan tidak adanya kontrol dinamis untuk mendeteksi keterlambatan pengembalian Barang Milik Daerah (BMD). Penelitian ini merancang aplikasi monitoring peminjaman aset berbasis web dengan fitur deteksi keterlambatan secara dinamis menggunakan metode Rapid Application Development (RAD). Sistem dikembangkan dengan Laravel 10, PHP 8.1+, dan MySQL 8.0, serta disimulasikan pada intranet lokal sesuai kebijakan keamanan data pemerintah. Kebaruan terletak pada mekanisme otomatisasi perubahan status 'Terlambat' berbasis kueri dinamis di layer aplikasi yang dievaluasi setiap modul diakses, bukan scheduler atau database trigger. Pendekatan ini dilengkapi dengan audit trail `tb_otomasi_keterlambatan` untuk menjamin akuntabilitas BMD sesuai Perda Kabupaten Magetan No. 1 Tahun 2021. Pengujian menggunakan Requirements Traceability Matrix (RTM) dan Blackbox Testing 19 skenario menunjukkan tingkat keberhasilan fungsional 100%. Sistem mampu mengubah proses manual menjadi real-time, memvalidasi stok otomatis, dan mencatat keterlambatan tanpa intervensi admin. Disimpulkan bahwa aplikasi layak diadopsi untuk digitalisasi tata kelola aset pemerintah daerah karena telah memenuhi kebutuhan fungsional, teknis, dan operasional.

**Kata Kunci**— Monitoring Peminjaman Aset, Kueri Dinamis, Rapid Application Development, Laravel, Blackbox Testing.

## I. PENDAHULUAN

Dinas Komunikasi dan Informatika (Diskominfo) Kabupaten Magetan memiliki peran sentral dalam mendukung tata kelola pemerintahan berbasis elektronik (e-Government) melalui pengelolaan dan penyediaan infrastruktur Teknologi Informasi dan Komunikasi (TIK). Untuk menjalankan tugas tersebut, Diskominfo bergantung pada sejumlah besar aset dan inventaris teknologi, seperti komputer, server, perangkat jaringan, proyektor, hingga peralatan video conference [1].

Pengelolaan aset daerah di Kabupaten Magetan diatur secara komprehensif dalam Peraturan Daerah No. 1 Tahun 2021 tentang Pengelolaan Barang Milik Daerah. Pasal 1 angka 8 mendefinisikan Barang Milik Daerah (BMD) sebagai semua barang yang dibeli atau diperoleh atas beban Anggaran Pendapatan dan Belanja Daerah atau berasal dari perolehan lainnya yang sah. Pasal 3 menegaskan bahwa tujuan pengelolaan BMD adalah untuk mewujudkan pengelolaan yang efektif dan efisien dalam mendukung penyelenggaraan urusan pemerintahan di daerah [2]. Dinas Komunikasi dan

Informatika Kabupaten Magetan adalah penyelenggara e-Government yang mengelola aset TIK seperti komputer, server, dan proyektor yang statusnya Barang Milik Daerah. Pengelolaan BMD diatur dalam Perda Kabupaten Magetan No. 1 Tahun 2021, di mana Pasal 1 angka 29 mendefinisikan 'Pinjam Pakai' sebagai penyerahan penggunaan barang tanpa imbalan dalam jangka waktu tertentu [2].

Deteksi keterlambatan secara dinamis merupakan penerapan konsep dynamic query dalam basis data temporal. Menurut Elmasri & Navathe (2016), dynamic SQL adalah teknik konstruksi kueri yang dieksekusi saat runtime berdasarkan nilai data terkini, sehingga memungkinkan pengambilan keputusan real-time tanpa perlu menyimpan hasil komputasi sebelumnya [3]. Berbeda dengan pendekatan scheduler atau database trigger, deteksi keterlambatan secara dinamis dilakukan melalui evaluasi kueri pada layer aplikasi setiap kali modul diakses. Pendekatan ini dipilih untuk menghindari ketergantungan pada konfigurasi trigger yang kompleks, sehingga sesuai dengan lingkungan server intranet Diskominfo yang memiliki alokasi sumber daya terbatas dan kebijakan keamanan ketat. Dalam perspektif Business Intelligence (BI) dan Decision Support Systems (DSS), keberadaan sistem monitoring berperan sebagai sarana pengawasan kondisi organisasi yang dilakukan secara berkelanjutan. Cakupannya meliputi fase pengenalan permasalahan, pemantauan kinerja operasional melalui indikator berbasis Key Performance Indicator (KPI), hingga penyediaan basis informasi bagi pengambil keputusan yang bersumber dari data historis maupun data terkini [4].

Berdasarkan observasi lapangan di Diskominfo Kabupaten Magetan pada Januari 2026, proses peminjaman aset TIK masih sepenuhnya konvensional (pengajuan menggunakan formulir fisik), persetujuan menunggu disposisi Kasubag Umum dan Kepegawaian (2–3 hari kerja), serta pencatatan stok menggunakan spreadsheet yang rentan konflik jadwal [1]. Saat ini, proses manajemen peminjaman aset di Diskominfo Kabupaten Magetan masih cenderung dilakukan secara konvensional [5].

Penelitian terdahulu oleh Oktaviani dkk. (2023) telah mengimplementasikan metode RAD untuk sistem peminjaman aset di lingkungan perusahaan swasta, serta merekomendasikan pengembangan lebih lanjut menuju sistem penyewaan aset dengan kompleksitas kebutuhan pengguna yang lebih tinggi [6]. Namun, implementasi di lingkungan pemerintahan daerah memerlukan penyesuaian fungsional khusus terkait akuntabilitas Barang Milik Daerah (BMD) dan mekanisme monitoring jatuh tempo yang belum terakomodasi.

Kusumojati dan Mediawati (2024) mengembangkan sistem informasi manajemen aset berbasis web di lingkungan perguruan tinggi dengan metode SDLC, namun tidak membahas mekanisme deteksi keterlambatan secara otomatis [7]. Implementasi di lingkungan pemerintahan daerah memerlukan penyesuaian fungsional yang berbeda, khususnya terkait akuntabilitas BMD, alur disposisi bertingkat, dan keamanan data intranet instansi.

Berdasarkan telaah komparatif, ketiadaan fitur pelacakan keterlambatan yang terintegrasi menjadi penyebab utama inefisiensi ketersediaan aset. Oleh karena itu, penelitian ini mengusulkan implementasi mekanisme deteksi keterlambatan yang dievaluasi secara dinamis setiap kali modul sistem diakses. Kebaruan berupa validasi status keterlambatan berbasis kueri dinamis pada layer aplikasi disesuaikan dengan prosedur pinjam pakai dan standar transparansi BMD. Pendekatan ini memungkinkan pembaruan status dari "Dipinjam" menjadi "Terlambat" secara real-time tanpa intervensi manual.

Rumusan masalah penelitian adalah: (1) Bagaimana menganalisis alur kerja manajemen peminjaman aset di Diskominfo Kabupaten Magetan yang masih konvensional untuk mengidentifikasi kebutuhan sistem yang terstruktur? (2) Bagaimana merancang dan mengimplementasikan aplikasi monitoring untuk mengonfirmasi keterlambatan? (3) Bagaimana memastikan sistem dapat menampilkan pengingat otomatis secara akurat saat status berubah? Penelitian bertujuan menjawab ketiga pertanyaan tersebut melalui:

1. Menganalisis alur kerja peminjaman, pengembalian, dan pelaporan aset yang berjalan secara konvensional di Diskominfo Kabupaten Magetan, serta mengidentifikasi kendala utama yang menyebabkan inefisiensi dan ketidakakuratan data.
2. Merancang dan mengimplementasikan aplikasi monitoring peminjaman aset untuk mengonfirmasi keterlambatan.
3. Menguji fitur pengingat otomatis sistem saat terjadi perubahan status peminjaman dari "Dipinjam" ke "Terlambat".

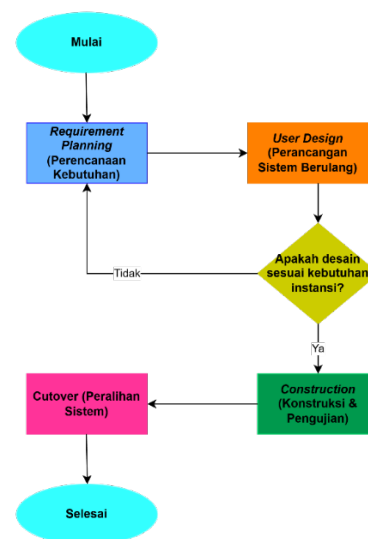
Sebagai batasan masalahnya, yaitu : 1) Sistem hanya mencakup tiga proses inti: peminjaman aset, pengembalian aset, dan pelaporan. Tidak mencakup pengadaan, penghapusan, atau integrasi dengan sistem keuangan daerah. 2) Pengguna dibatasi pada tiga level hak akses: Administrator, Peminjam, dan Verifikator. 3) Implementasi menggunakan PHP 8+ dengan framework Laravel 10 diuji pada lingkungan server lokal XAMPP. Sistem tidak dikembangkan dalam bentuk aplikasi mobile Android/iOS atau desktop.

## II. METODE PENELITIAN

### A. Metode Pengembangan

Penelitian ini mengadopsi metode *Rapid Application Development* (RAD) menurut James Martin (1991) yang terdiri dari empat tahapan yang terstruktur: *Requirement Planning*, *User Design*, *Construction*, dan *Cutover* [8]. RAD dipilih karena mendukung siklus pengembangan singkat

melalui keterlibatan aktif pengguna secara berulang, memotong birokrasi pengembangan konvensional, dan memungkinkan validasi prototipe secara iteratif sebelum konstruksi final [9].



Gbr 1 Alur Penelitian Metode RAD [8]

### B. Pengumpulan Data dan Analisis Kebutuhan

Teknik pengumpulan data yang digunakan dalam penelitian ini bersifat kombinasi (triangulasi) untuk menjamin data yang diperoleh lengkap, akurat, dan sesuai dengan kebutuhan pengembangan aplikasi. Teknik tersebut meliputi:

- Observasi: Dilakukan untuk mengamati dan memahami secara langsung lingkungan kerja, proses bisnis eksisting, dan objek fisik yang akan dikelola oleh sistem. Prosedur observasi lingkungan dilakukan dengan mengamati alur kerja harian di Diskominfo, khususnya petugas yang bertanggung jawab mengelola aset.
- Wawancara: Digunakan untuk menggali informasi mendalam mengenai kebutuhan fungsional sistem dan mendapatkan validasi terhadap prosedur dan aturan yang berlaku. Pengurus Barang sebagai *end-user* dan pelaksana teknis, serta Kasubag Umum dan Kepegawaian sebagai *decision maker*, diwawancarai mengenai kebijakan manajemen aset, aturan baku batas waktu pengembalian, dan sanksi yang harus diakomodasi oleh sistem.
- Studi pustaka: Digunakan untuk mengumpulkan data teoretis, konseptual, dan hasil penelitian terdahulu yang relevan sebagai landasan teoretis dan justifikasi teknis/metodologis.
- Analisis Dokumen: Digunakan untuk memperoleh data pendukung formal sebagai input awal dalam pengisian

database berupa data inventaris aset konvensional terbaru.

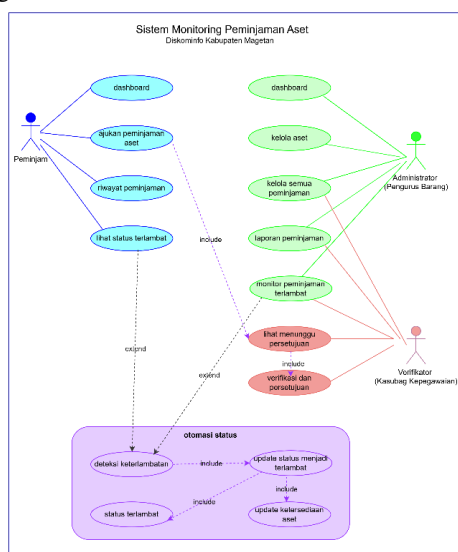
### C. Perancangan Sistem

Berdasarkan hasil triangulasi data, kebutuhan sistem dikelompokkan menjadi dua kategori utama:

- Fungsional: Otentikasi berbasis peran, CRUD aset TIK, alur peminjaman digital dengan validasi stok *real-time*, disposisi verifikator, pencatatan pengembalian, otomasi status keterlambatan, beranda monitoring, dan ekspor laporan PDF.
- Non-Fungsional: Keamanan (*bcrypt*, CSRF), kinerja respons <5 detik, keandalan ACID MySQL 8.0, kompatibilitas dengan browser modern, dan kepatuhan terhadap Perda Kabupaten Magetan No. 1/2021.

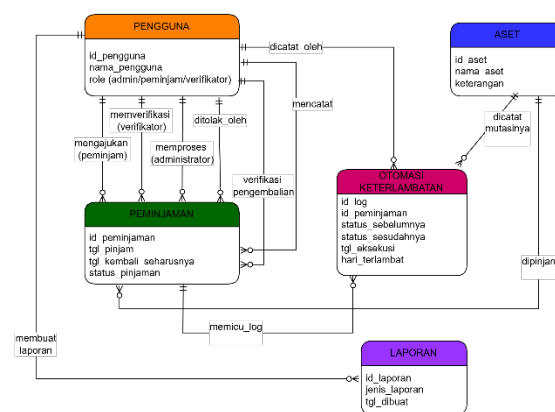
Berdasarkan hasil analisis kebutuhan, interaksi aktor dengan sistem dipetakan menggunakan *Use Case Diagram*. Diagram ini mendefinisikan batasan sistem (system boundary) dan hak akses berbasis peran:

- Peminjam mengajukan pinjaman, memantau riwayat, dan melihat status keterlambatan.
- Administrator mengelola data aset (CRUD), mencatat serah terima fisik, dan mencetak laporan.
- Verifikator memberikan disposisi (setuju/tolak), memantau keterlambatan, dan memverifikasi kondisi pengembalian.



Gbr 2 Use Case Diagram

Untuk merepresentasikan struktur data tersebut secara konseptual, dikembangkan *Conceptual Data Model* (CDM). CDM memetakan entitas-entitas seperti PENGGUNA, ASET, PEMINJAMAN, OTOMASI KETERLAMBATAN, dan LAPORAN beserta relasi antar-entitas sesuai aturan bisnis pengelolaan aset pemerintah daerah.

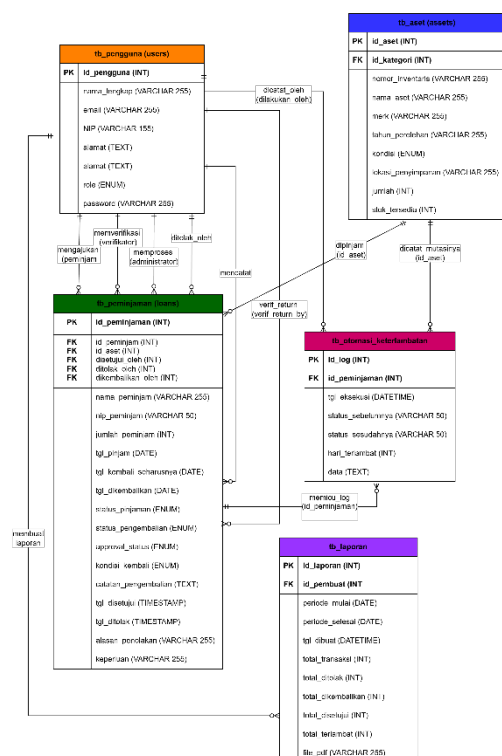


Gbr 3 Conceptual Data Model (CDM)

Entitas PENGGUNA dan relasinya merepresentasikan seluruh aktor (administrator, peminjam, dan verifikator) yang memiliki karakteristik dasar yang sama (identitas, kontak), sehingga efisien dikelola dalam satu struktur. Entitas ASET merepresentasikan Barang Milik Daerah berbasis TIK dengan atribut konseptual: identitas, spesifikasi fisik, kondisi, dan lokasi. Entitas transaksional yang merekam setiap peristiwa pinjam-pakai adalah PEMINJAMAN. Entitas ini merupakan transaksi yang menghubungkan peminjam dengan aset yang dipinjam dan mencatat siklus hidup peminjaman dari pengajuan hingga pengembalian. Entitas khusus untuk audit trail dan sistem yang mendeteksi keterlambatan secara otomatis adalah OTOMASI KETERLAMBATAN. Ini merupakan implementasi novelty untuk transparansi monitoring. Entitas LAPORAN merupakan rekapitulasi aktivitas hasil olahan data untuk dipertanggungjawabkan.

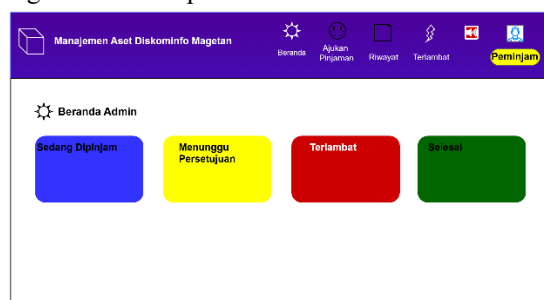
Setelah model konseptual divalidasi oleh Pengurus Barang serta Kasubag Umum dan Kepegawaian, CDM diturunkan menjadi *Physical Data Model* (PDM), seperti yang terlihat pada Gbr.4. Arsitektur basis data terdiri atas enam tabel utama yang saling terelasi. Tabel *tb\_pengguna* mengakomodasi tiga peran sistem melalui tipe data ENUM, menyimpan kredensial terenkripsi, dan terhubung ke tabel transaksi untuk pelacakan audit berbasis kapasitas peran. Tabel *tb\_aset* merepresentasikan inventaris BMD TIK dengan logika bisnis dinamis yang secara otomatis mengurangi *stok\_tersedia* saat persetujuan dan menambahkannya kembali saat pengembalian tervalidasi. Tabel transaksi inti *tb\_peminjaman* mencatat siklus hidup peminjaman dan memuat tiga *Foreign Key* ke *tb\_pengguna* (penyetuju, penolak, dan petugas pengembali) guna menjamin transparansi jejak audit. Kebaruan (*novelty*) penelitian diimplementasikan pada tabel *tb\_otomasi\_keterlambatan*, yang berfungsi sebagai *audit trail* sistem deteksi keterlambatan otomatis; tabel ini mencatat *timestamp* eksekusi, perubahan status, serta durasi keterlambatan tanpa intervensi konvensional [10]. Seluruh struktur dikembangkan menggunakan Laravel 10 dan PHP 8.1+, dilengkapi penjadwalan dengan sistem kueri dinamis untuk validasi jatuh tempo, serta tabel *tb\_laporan* yang

menyimpan metadata rekapitulasi untuk optimasi pengambilan data historis.



Gbr 4 Physical Data Model (PDM)

Setelah pemodelan bisnis dan data selesai dirancang, tahap selanjutnya dalam fase User Design adalah pembuatan prototipe antarmuka (*mockup*). Dalam metodologi *Rapid Application Development* (RAD), prototipe berfungsi sebagai representasi visual dari sistem yang akan dibangun, sehingga pengguna akhir dapat memberikan umpan balik secara langsung sebelum tahap konstruksi dimulai.



Gbr 5 Mockup Status Keterlambatan

Status keterlambatan mockup ditandai dengan badge merah pada masing-masing beranda peminjam, verifikator, dan administrator. Status tersebut muncul setelah ada perubahan jika peminjaman melebihi batas waktu *tgl\_dikembalikan\_sharusnya*, seperti pada Gbr. 5.

#### D. Konstruksi dan Implementasi

Tahap konstruksi dilaksanakan menggunakan Laravel 10 dengan PHP 8.1+ dan MySQL 8.0 pada lingkungan lokal

XAMPP. Arsitektur MVC diterapkan dengan pemisahan *model*, *controller*, dan *view* yang jelas. *Frontend* dibangun menggunakan *Blade Templating Engine* dan *Bootstrap 5.3.8*. *Novelty* sistem diimplementasikan dengan mengeksekusi kueri UPDATE secara dinamis setiap kali modul diakses, menggantikan pendekatan *database trigger* yang bergantung pada konfigurasi DBMS-level [11]. Seluruh transaksi basis data dijamin dengan properti ACID MySQL 8.0, sedangkan keamanan akses menggunakan enkripsi *bcrypt* dan proteksi CSRF bawaan *Laravel* [12].

#### E. Validasi dan Pengujian Sistem

Data yang diperoleh dari observasi, wawancara, dan dokumen dianalisis menggunakan triangulasi metode untuk memastikan konsistensi antarsumber. Temuan kualitatif ditransformasikan ke dalam spesifikasi teknis melalui pemetaan kebutuhan fungsional dan nonfungsional. Selanjutnya, validasi kualitas sistem dilaksanakan pada fase *Cutover* melalui dua instrumen utama:

##### 1) Requirements Traceability Matrix (RTM)

RTM memetakan linearitas antara kebutuhan bisnis, desain teknis, implementasi modul, dan skenario uji Blackbox untuk menjamin tidak ada fitur yang terlewat.

##### 2) Blackbox Testing

Validasi fungsionalitas tanpa meninjau kode sumber, berfokus pada kesesuaian *output* sistem terhadap *input* yang diberikan berdasarkan spesifikasi kebutuhan yang telah ditetapkan.

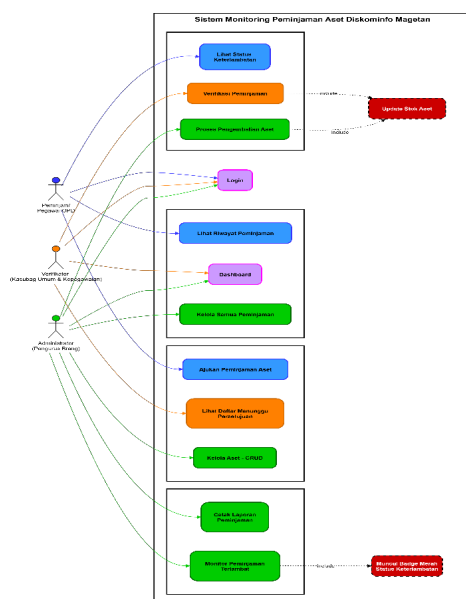
### III. HASIL DAN PEMBAHASAN

#### A. Hasil Perencanaan Kebutuhan

Berdasarkan observasi, alur konvensional menyebabkan proses peminjaman lambat karena menunggu persetujuan fisik dari Kasubag Umum dan Kepegawaian. Hasil wawancara dengan Pengurus Barang menunjukkan kesulitan melacak stok sdet secara *real-time* dan sering terjadi konflik jadwal peminjaman aset yang sama. Tidak ada mekanisme otomatis untuk menagih aset yang terlambat dikembalikan. Hal ini menimbulkan inefisiensi waktu, kesalahan stok, dan tidak adanya otomatisasi status keterlambatan.

*Use Case Diagram* menghasilkan struktur basis data yang menempatkan *tb\_peminjaman* sebagai tabel transaksi utama yang menghubungkan *tb\_user*, *tb\_aset* dan *tb\_otomasi\_keterlambatan*. Tabel *tb\_otomasi\_keterlambatan* menjadi kunci untuk menyimpan logika validasi keterlambatan. *Use case* yang melibatkan aktor ini meliputi: peminjam, verifikator dan administrator. *Peminjam* login ke dalam sistem, mengajukan peminjaman aset, melihat riwayat peminjaman untuk memantau status pengajuan yang telah dibuat, serta melihat status keterlambatan apabila terdapat aset yang belum dikembalikan melewati batas waktu. *Verifikator*

berperan sebagai pihak yang berwenang melakukan verifikasi dan persetujuan terhadap pengajuan peminjaman. Proses verifikasi ini memiliki relasi include dengan use case Update Stok Aset, karena setiap peminjaman yang disetujui akan otomatis mengurangi ketersediaan stok aset di dalam sistem. *Administrator* adalah pengelola utama sistem yang memiliki hak akses paling luas. Administrator dapat mengakses Dashboard, Kelola Aset - CRUD untuk menambah, mengubah, dan menghapus data aset, Kelola Semua Peminjaman untuk melihat seluruh transaksi, Proses Pengembalian Aset saat barang dikembalikan oleh peminjam, Monitor Peminjaman Terlambat untuk mengidentifikasi keterlambatan, dan Cetak Laporan Sementara, Monitor Peminjaman Terlambat memiliki relasi include dengan memunculkan badge merah sebagai tanda status keterlambatan yang akan mengirimkan peringatan otomatis kepada peminjam.



Gbr 6 Use CAs Diagram Monitoring Peminjaman Aset Diskominfo Kabupaten Magetan

### B. Hasil Validasi Kebutuhan RTM

Untuk menjamin keterlacakan (traceability) antara kebutuhan operasional Diskominfo dengan implementasi teknis, penelitian menyusun RTM sebagaimana disajikan pada Tabel I. Matriks ini memetakan 7 kebutuhan fungsional (REQ-01 s.d. REQ-07) dan 2 kebutuhan non-fungsional (NFR-01, NFR-02) ke dalam spesifikasi modul, diagram use case, serta ID kasus uji.

TABEL I  
HASIL VALIDASI DENGAN RTM

| ID   | Deskripsi Persyaratan | Kategori   | Modul Aplikasi | ID Kasus Uji | Status   |
|------|-----------------------|------------|----------------|--------------|----------|
| REQ- | Otentikasi            | Fungsional | Login/         | BB-01        | Berhasil |

| ID     | Deskripsi Persyaratan                                    | Kategori                 | Modul Aplikasi                  | ID Kasus Uji           | Status   |
|--------|----------------------------------------------------------|--------------------------|---------------------------------|------------------------|----------|
| 01     | aman & routing peran                                     | & Keamanan               | Logout, Middleware              | (Skenario 1-5)         |          |
| REQ-02 | CRUD aset & validasi unik inventaris                     | Fungsional               | Master Data Aset                | BB-02 (Skenario 6-9)   | Berhasil |
| REQ-03 | Pengajuan digital & cek stok <i>real-time</i>            | Fungsional               | Ajukan Peminjaman               | BB-03 (Skenario 10,13) | Berhasil |
| REQ-04 | Disposisi verifikasi & cegah manipulasi stok             | Fungsional               | Verifikasi & Persetujuan        | BB-04 (Skenario 11-12) | Berhasil |
| REQ-05 | Pencatatan pengembalian & kembalikan stok                | Fungsional               | Input & Verifikasi Pengembalian | BB-05 (Skenario 14-17) | Berhasil |
| REQ-06 | Otomasi status 'Terlambat' tanpa intervensi konvensional | Fungsional & Kinerja     | Modul Otomasi (Kueri Dinamis)   | BB-06 (Skenario 18)    | Berhasil |
| REQ-07 | Ekspor rekap transaksi per periode ke PDF                | Fungsional               | Export Laporan PDF              | BB-07 (Skenario 19)    | Berhasil |
| NFR-01 | Berjalan di XAMPP (PHP 8.1+) intranet                    | Kinerja & Kompatibilitas | Environment Laravel 10          | NFR-01                 | Berhasil |
| NFR-02 | Waktu respons kueri monitoring < 3 detik                 | Kinerja                  | Optimasi Query MySQL            | NFR-02                 | Berhasil |

Hasil Tabel I menunjukkan seluruh butir kebutuhan terpetakan utuh dan berstatus berhasil. Validasi NFR-01 membuktikan bahwa arsitektur selaras dengan kebijakan isolasi BMD, sementara NFR-02 mengonfirmasi efisiensi pendekatan kueri dinamis dibandingkan dengan mekanisme *trigger* berat.

### C. Hasil Pengujian Blackbox

Pengujian Blackbox divalidasi terhadap 19 skenario fungsional. Hasil ringkas disajikan pada Tabel II.

TABEL II  
HASIL PENGUJIAN BLACKBOX

| No    | Modul                 | Skenario Uji                    | Hasil                           | Status   |
|-------|-----------------------|---------------------------------|---------------------------------|----------|
| 1-5   | Hak Akses             | Login/Logout 3 <i>roles</i>     | Sesuai ekspektasi               | Berhasil |
| 6-9   | Manajemen Aset        | CRUD & Validasi Unik            | Data tersimpan/ terupdate       | Berhasil |
| 10-13 | Peminjaman            | Ajukan/Setujui/Tolak/Stok Habis | Status terupdate, stok valid    | Berhasil |
| 14-17 | Pengembalian          | Input Kondisi & Verifikasi      | Stok bertambah, riwayat selesai | Berhasil |
| 18    | Otomasi Keterlambatan | Auto-update status jatuh tempo  | Status berubah otomatis         | Berhasil |

|    |         |            |                |          |
|----|---------|------------|----------------|----------|
| 19 | Laporan | Export PDF | File terbentuk | Berhasil |
|----|---------|------------|----------------|----------|

Seluruh modul berjalan sesuai spesifikasi. Validasi input pada skenario stok habis (No. 13) membuktikan mekanisme *auto-lock* efektif mencegah konflik jadwal. Skenario nomor 18 mengonfirmasi kebaruan berjalan secara konsisten melalui mekanisme evaluasi kueri terintegrasi.

#### D. Hasil Simulasi Peralihan Terbatas (Cutover)

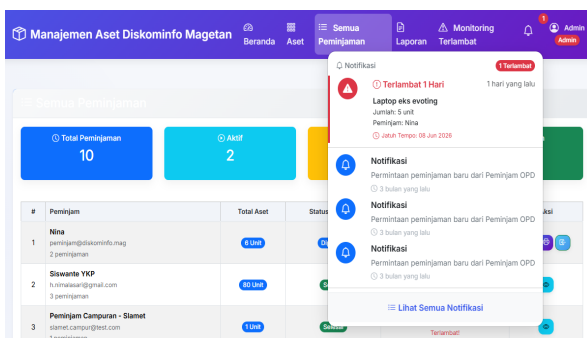
Karena penelitian dibatasi pada skala simulasi demi memenuhi regulasi keamanan data BMD, fase *cutover* menghasilkan prototipe siap pakai pada lingkungan kerja lokal terbatas. Keluaran nyata meliputi:

1. Konfigurasi Server Lokal: Aplikasi berhasil dikonfigurasi dan dijalankan pada Apache (XAMPP) di komputer uji coba pengurus barang.
2. Migrasi Sampel Data Riil: Basis data MySQL diisi dengan sampel inventaris aset TIK asli untuk menguji ketahanan kueri dinamis terhadap data operasional.
3. Demonstrasi Operasional: Peragaan sistem memastikan pengguna memahami arti perubahan warna otomatis *badge* status 'Terlambat' pada beranda aplikasi.

#### E. Pemetaan Hasil terhadap Tujuan Penelitian

Menjawab tujuan penelitian 1 (Digitalisasi Alur Manual Menjadi Kebutuhan Terstruktur): Sistem mengimplementasikan arsitektur 3 role dengan validasi stok dinamis. Keterlacakan kebutuhan terverifikasi penuh pada RTM dan tervalidasi melalui RTM REQ-01 s.d. REQ-05 dan REQ-07. Hal ini juga dibuktikan pada skenario 1-17 *Blackbox Testing*.

Menjawab tujuan penelitian 2 (Implementasi Monitoring keterlambatan Berbasis Kueri Dinamis): Aplikasi berhasil memberi informasi status keterlambatan melalui evaluasi kueri dinamis yang terintegrasi pada akses modul. Berbeda dengan permasalahan sebelumnya yang mengandalkan pengecekan manual, pendekatan ini lebih ringan sesuai dengan infrastruktur intranet Diskominfo. Keberhasilan ini dibuktikan pada RTM REQ-06 dan *Blackbox Testing skenario 18*.



Gbr 7 Badge merah status keterlambatan pada Beranda Administrator

Menjawab tujuan penelitian 3 (Akurasi Peningkat Otomatis Perubahan Status): Pengujian Blackbox skenario 18 membuktikan sistem secara akurat mengubah status dari "Dipinjam" menjadi "Terlambat" saat melewati jatuh tempo, mencatat eksekusi ke `tb_otomasi_keterlambatan` dengan timestamp, dan menampilkan indikator visual badge merah di beranda intervensi admin (Gbr.7).

#### F. Pembahasan dan Analisis Kebaruan

Kebaruan penelitian ini terletak pada implementasi mekanisme deteksi keterlambatan yang dievaluasi secara dinamis setiap kali modul sistem diakses, bukan menggunakan database *trigger* atau *cron job*.

```
private function updateStatusTerlambat()
{
    // 1. Ambil semua loan yang harusnya udah terlambat
    $loansTerlambat = Loan::where('status_pinjaman', 'Dipinjam') // atau 'Disetujui'
        ->where('tgl_kembali_seharusnya', '<', now())
        ->whereNull('tgl_dikembalikan')
        ->get();

    foreach ($loansTerlambat as $loan) {
        DB::beginTransaction();
        try {
            // 2. Update status di tabel loans
            $statusSebelum = $loan->status_pinjaman;
            $loan->status_pinjaman = 'Terlambat';
            $loan->save();

            // 3. INSERT KE TB_OTOMASI_KETERLAMBATAN
            DB::table('tb_otomasi_keterlambatan')->insert([
                'loan_id' => $loan->id,
                'status_sebelum' => $statusSebelum, // 'Dipinjam'
                'status_sesudah' => 'Terlambat',
                'waktu_ubah' => now(),
                'created_at' => now(),
                'updated_at' => now(),
            ]);

            DB::commit();

            \Log::info('Otomasi keterlambatan berhasil', [
                'loan_id' => $loan->id,
                'nama_peminjam' => $loan->nama_peminjam
            ]);
        } catch (\Exception $e) {
            DB::rollBack();
            \Log::error('Gagal update status terlambat: ' . $e->getMessage());
        }
    }
}
```

Gbr 8 Implementasi Method `updateStatusTerlambat` di `LoanController.php`

Mekanisme kebaruan berupa deteksi keterlambatan berbasis kueri dinamis yang dilengkapi *audit trail* `tb_otomasi_keterlambatan` untuk menjamin akuntabilitas Barang Milik Daerah Kabupaten Magetan. Gbr.6 menampilkan method `updateStatusTerlambat()` pada `LoanController.php` yang menjadi inti dari fitur otomasi keterlambatan di REQ-06. Terdapat 3 bagian utama dalam method ini:

1. Kueri dinamis `WHERE tgl_kembali < CURRENT_DATE`  
`$loansTerlambat= Loan::where('status_pinjaman', 'Dipinjam')`  
`->where('tgl_kembali_seharusnya', '<', now())`  
`//`

```
->whereNull('tgl_dikembalikan')
->get();
```

## 2. Update status otomatis

Setelah data terseleksi, sistem melakukan perulangan *foreach* untuk mengubah status\_pinjaman dari 'Dipinjam' menjadi 'Terlambat' menggunakan `$loan->save()`. Proses ini berjalan tanpa intervensi admin.

## 3. Pencatatan audit trail ke tb\_otomasi\_keterlambatan

```
DB::table('tb_otomasi_keterlambatan')->insert([
    'loan_id' => $loan->id,
    'status_sebelum' => $statusSebelum,
    'status_sesudah' => 'Terlambat',
    'waktu_ubah' => now(),
]);
```

Setiap kali pengguna mengakses *dashboard* atau modul, sistem secara otomatis:

1. Mendeteksi peminjaman yang melewati jatuh tempo
2. Mengubah status dari "Dipinjam" menjadi "Terlambat"
3. Sistem berhasil mencatat log perubahan status pada tabel `tb_otomasi_keterlambatan`. Mencatat eksekusi ke `tb_otomasi_keterlambatan` dengan timestamp (Gbr.9)
4. Menghitung jumlah hari keterlambatan
5. Menampilkan indikator visual (badge merah) di beranda.

| loan_id | status_sebelum | status_sesudah | waktu_ubah          | created_at          | updated_at          |
|---------|----------------|----------------|---------------------|---------------------|---------------------|
| 1       | Dipinjam       | Terlambat      | 2026-06-23 06:07:14 | 2026-06-23 06:07:14 | 2026-06-23 06:07:14 |
| 45      | Dipinjam       | Terlambat      | 2026-06-23 06:07:14 | 2026-06-23 06:07:14 | 2026-06-23 06:07:14 |

Gbr 9 Tampilan `tb_otomasi_keterlambatan`

Tabel III

STRUKTUR TABEL `TB_OTOMASI_KETERLAMBATAN`

| Field                       | Tipe Data   | Keterangan                      |
|-----------------------------|-------------|---------------------------------|
| <code>id_log</code>         | BIGINT      | Primary Key                     |
| <code>id_peminjaman</code>  | BIGINT      | FK → <code>tb_peminjaman</code> |
| <code>Waktu_ubah</code>     | DATETIME    | Timestamp eksekusi              |
| <code>status_sebelum</code> | VARCHAR(50) | Status sebelum                  |
| <code>status_sesudah</code> | VARCHAR(50) | Status sesudah                  |
| <code>hari_terlambat</code> | INT         | Jumlah hari                     |

Pendekatan ini dipilih untuk menghindari ketergantungan pada konfigurasi *trigger* tingkat basis data yang kompleks, sehingga lebih sesuai dengan lingkungan server intranet Diskominfo yang memiliki alokasi sumber daya terbatas dan kebijakan keamanan yang ketat.

Hasil perbandingan antara kondisi eksisting dan sistem digital disajikan pada Tabel IV.

Tabel IV

PERBANDINGAN SISTEM KONVENSIONAL VS SISTEM DIGITAL

| Aspek | Sistem Konvensional | Sistem Digital | Peningkatan |
|-------|---------------------|----------------|-------------|
|-------|---------------------|----------------|-------------|

| Aspek                    | Sistem Konvensional                            | Sistem Digital                             | Peningkatan |
|--------------------------|------------------------------------------------|--------------------------------------------|-------------|
| Waktu Persetujuan        | 2–3 hari kerja (disposisi fisik)               | < 3 detik (real-time digital)              | Signifikan  |
| Pelacakan Stok           | Spreadsheet konvensional, rawan konflik jadwal | Real-time dengan auto-lock saat stok habis | Signifikan  |
| Monitoring Keterlambatan | Pengecekan konvensional satu per satu          | Otomatis setiap kali server aktif          | Signifikan  |
| Pelaporan                | Rekap konvensional > 3 hari                    | Auto-generate PDF < 5 menit                | Signifikan  |

Sistem menyediakan tiga perspektif beranda yang saling terintegrasi sesuai dengan *role* pengguna:

- Beranda Administrator: Menampilkan menu cepat untuk akses langsung ke fitur "Tambah Aset", "Semua Peminjaman", "Cetak Laporan", dan "Monitoring Terlambat" seperti terlihat pada Gbr. 10.

| No. Inventaris | Nama Aset          | Merk   | Kondisi     | Lokasi  | Jumlah | Stok Tersedia | Status       | Aksi       |
|----------------|--------------------|--------|-------------|---------|--------|---------------|--------------|------------|
| 001            | Laptop eks evoting | Lenovo | baik        | Magetan | 150    | 150           | 150 Tersedia | Edit Hapus |
| 002            | Kamera             | Canon  | Rusak       | Magetan | 2      | 2             | 2 Tersedia   | Edit Hapus |
| 003            | Laptop eks evoting | Lenovo | rusak berat | Magetan | 1      | 1             | 1 Tersedia   | Edit Hapus |

Gbr 10 Beranda Administrator

- Beranda Verifikator: Gbr. 11 berfokus pada menu persetujuan dengan daftar peminjaman yang menunggu. Verifikator dapat melihat detail keperluan, jangka waktu, dan identitas peminjam sebelum memutuskan untuk menyetujui atau menolak.

| No | Peminjam                            | Aset               | Jumlah | Tgl Kembali | Kondisi | Status  | Aksi       |
|----|-------------------------------------|--------------------|--------|-------------|---------|---------|------------|
| 1  | Peminjam OPD peminjam@diskominfo.mg | Laptop eks evoting | 150    | 26 Apr 2026 | 150     | 2 Menit | Verifikasi |

Gbr 11 Beranda Verifikator

- Formulir Peminjam: Dirancang dengan prinsip *User-Centered Design* untuk memudahkan staf dalam mengajukan dan melihat status aset. Formulir ini

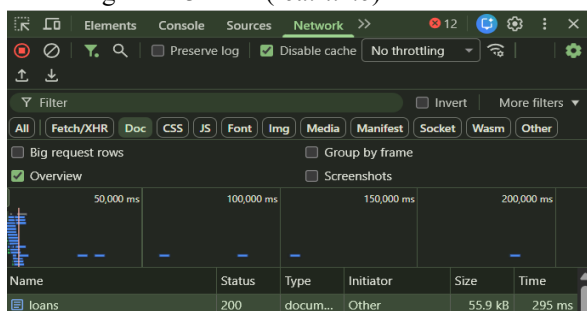
memungkinkan peminjam mengisi data diri, nama, NIP, jabatan, alamat, dan detail aset yang diinginkan (Gbr.12).

Gbr 12 Formulir Peminjam

Implementasi aplikasi monitoring peminjaman aset telah mengubah proses konvensional menjadi sistem digital yang terotomasi dengan peningkatan signifikan:

Waktu Persetujuan:

- Sistem konvensional: waktu tunggu lama (2-3 hari) karena menunggu disposisi fisik (Observasi, Januari 2026)
- Sistem digital: < 3 detik (*real-time*)



Gbr 13 Pengujian Performa NFR-02 Semua Peminjaman

Waktu respons 295 ms pada Gbr.11 membuktikan sistem memenuhi kriteria NFR-02, yaitu waktu *respons query monitoring* < 3 detik. Pengujian dilakukan dengan 47 data peminjaman pada environment XAMPP PHP 8.1 secara lokal di jaringan intranet Diskominfo.

Pelacakan Stok:

- Sistem konvensional: Spreadsheet, sering konflik jadwal
- Sistem digital: *Real-time* dengan *auto-lock*
- Mencegah *bottleneck* aset vital

Monitoring Keterlambatan:

- Sistem konvensional: Pengecekan konvensional satu per satu
- Sistem digital: Otomatis 24 jam selama server aktif
- Tidak ada peminjaman yang terlewatkan

Pelaporan:

- Sistem konvensional: lama bisa lebih 3 hari untuk rekapan
- Sistem digital: < 5 menit (auto-generate PDF)

Hasil penelitian langsung menjawab ketiga rumusan masalah. Proses konvensional yang memakan waktu 2-3 hari

berhasil dipetakan menjadi 7 kebutuhan fungsional terstruktur (REQ-01 s.d. REQ-07) dan validasi penuh melalui RTM serta 19 skenario Blackbox awal. Aplikasi berhasil mengonfirmasi status keterlambatan melalui evaluasi kueri dinamis *tgl\_kembali\_seharusnya* < *CURRENT\_DATE* yang terintegrasi pada akses modul. Berbeda dengan penelitian terdahulu yang mengandalkan pengecekan konvensional, pendekatan ini lebih ringan dan sesuai dengan infrastruktur intranet Diskominfo. Hasil pengujian menunjukkan bahwa implementasi sistem telah memenuhi kriteria fungsional dan nonfungsional yang ditetapkan.

Keberhasilan skenario otomasi status (Tabel II, No. 18) menunjukkan bahwa sistem berhasil mendeteksi transaksi melewati jatuh tempo, memperbarui status\_pinjam menjadi "Terlambat", dan mencatat eksekusi ke *tb\_otomasi\_keterlambatan* tanpa memicu duplikasi data atau *error*. Pendekatan ini mengurangi beban I/O basis data, menghindari ketergantungan pada konfigurasi *cron* kompleks, dan tetap memenuhi prinsip akuntabilitas BMD melalui pencatatan timestamp, status sebelum/sesudah, serta jumlah hari keterlambatan.

Ketika sumber daya server terbatas, meminimalkan ketergantungan logika bisnis pada level DBMS sangat penting. Hal ini sesuai dengan prinsip integritas relasional yang menekankan efisiensi tersebut.

Dengan demikian, sistem ini bukan sekadar alat pencatatan, melainkan instrumen pendukung keputusan (*decision support*) yang memungkinkan pengurus barang dan verifikator melakukan intervensi administratif secara proaktif.

#### IV. KESIMPULAN

Berdasarkan hasil penelitian dan pembahasan, dapat disimpulkan bahwa:

1. Analisis alur kerja konvensional di Diskominfo Kabupaten Magetan berhasil diidentifikasi dan ditransformasikan menjadi 7 kebutuhan fungsional terstruktur.
2. Aplikasi monitoring berhasil dirancang dan diimplementasikan menggunakan Laravel 10 dengan mekanisme kebaruan berupa deteksi keterlambatan berbasis kueri dinamis. Fitur ini mengonfirmasi status aset yang melewati jatuh tempo secara otomatis tanpa intervensi manusia, sesuai dengan infrastruktur server lokal instansi.
3. Pengujian fungsional (19 skenario *Blackbox*) dan matriks keterlacakan (RTM) membuktikan sistem berjalan akurat dan konsisten. Perubahan status ditampilkan secara *real-time* pada beranda, disertai pencatatan *audit trail* (*tb\_otomasi\_keterlambatan*) yang

menjamin transparansi dan kepatuhan terhadap Perda Kabupaten Magetan Nomor 1 Tahun 2021.

4. Sistem telah memenuhi tujuan penelitian; keterbatasan membuka peluang penelitian selanjutnya.

#### REFERENSI

- [1] Dinas Komunikasi dan Informatika Kabupaten Magetan, *SOP Pelayanan Permintaan Pengadaan Aplikasi dari OPD ke Dinas Kominfo*, <https://kominfo.magetan.go.id/>. Accessed: Oct. 24, 2025. [Online]. Available: <https://kominfo.magetan.go.id/profil-dinas/>
- [2] Pemerintah Kabupaten Magetan, *Peraturan Daerah Kabupaten Magetan Nomor 1 Tahun 2021 tentang Pengelolaan Barang Milik Daerah*, p. 6, 2021.
- [3] S. B. Elmasri, Ramez, and Navathe, *Fundamentals of Database Systems*, seventh edition. 2016.
- [4] R. H. R. and I. Sugiri, *Business intelligence and decision support*, pp. 322–358, 2015.
- [5] Diskominfo Kab. Magetan, *Sop Pelayanan Permintaan Pengadaan Aplikasi Dari Opd Ke Dinas Kominfo*, 2021, [Online]. Available: <https://kominfo.magetan.go.id/sop-pelayanan-permintaan-pengadaan-aplikasi-dari-opd-ke-dinas-kominfo/>
- [6] Yohana Christela Oktaviani, Yulia Wahyuningsih, and Fernandi Mahendrasusila, *Penerapan rapid application development dalam rancang bangun sistem peminjaman aset PT Ukirama Solusi Indonesia*, INFOTECH J. Inform. Teknol., vol. 4, no. 2, pp. 292–305, 2023, doi: 10.37373/infotech.v4i2.911.
- [7] A. Wahab, J. Andini, R. Ashara, and W. Haryono, *Implementasi Metode RAD Pada Perancangan Aplikasi Perpustakaan Berbasis Web (Studi Kasus : SMKN 12 Kabupaten Tangerang)*, OKTAL J. Ilmu Komput. dan Sci. , vol. 2, no. 8, 2023.
- [8] J. Martin, *Rapid Application Development*, 3rd ed., vol. 788. Macmillan Publishing Company, 1991.
- [9] D. D. E. Turban, R. Sharda, *Decision Support and Business Intelligence Systems*, 9th ed. 2011.
- [10] D. A. Anggoro, W. Supriyanti, and D. A. P. Putri, *Konsep Dasar Sistem Basis Data dengan MySQL* - Google Books. 2021. [Online]. Available: [https://www.google.co.id/books/edition/Konsep\\_Dasar\\_Sistem\\_Basis\\_Data\\_dengan\\_My/xpFUEAAQBAJ?hl=id&gbpv=1&dq=Konsep+da](https://www.google.co.id/books/edition/Konsep_Dasar_Sistem_Basis_Data_dengan_My/xpFUEAAQBAJ?hl=id&gbpv=1&dq=Konsep+dasar+sistem+basis+data+dengan+mysql&pg=PR3&printsec=frontcover)
- [11] E. L. P. A. Rozaq; and R. N. Pratama, *Pemrograman Web Lanjutan Dengan Laravel*, Cetakan Pe. 2024.
- [12] Laravel LLC, *Laravel 10.x Documentation: Task Scheduling & Query Optimization*, 2024, [Online]. Available: <https://laravel.com/docs/10.x>