
Implementasi Metode Scrumban untuk Perancangan Ulang Sistem Informasi dan Manajemen Laboratorium (SIMLAB) Institut Teknologi Kalimantan

Aidil Saputra Kirsan*¹, Happy Aprillia², Adi Mahmud Jaya Marindra³

¹) Sistem Informasi, Teknik Elektro Informatika dan Bisnis, Institut Teknologi Kalimantan

^{2,3}) Teknik Elektro, Teknik Elektro Informatika dan Bisnis, Institut Teknologi Kalimantan

Email: *¹aidil@lecturer.itk.ac.id, ²happy.aprillia@lecturer.itk.ac.id, ³adi.marindra@lecturer.itk.ac.id

(Naskah masuk: 28 Oktober 2025, diterima untuk diterbitkan: 15 April 2026)

Abstrak: Sistem Informasi dan Manajemen Laboratorium (SIMLAB) merupakan solusi digitalisasi yang dirancang untuk meningkatkan efisiensi pengelolaan laboratorium di Institut Teknologi Kalimantan. Studi ini mengadopsi metode Scrumban, yaitu gabungan antara Scrum dan Kanban, sebagai pendekatan hybrid yang memberikan fleksibilitas dalam pengelolaan backlog sekaligus menjaga struktur kerja yang terorganisir melalui sprint. Pengembangan sistem dilakukan menggunakan teknologi Laravel untuk backend dan React untuk frontend dengan penerapan design pattern Repository, Service, Observer, dan Factory guna mendukung arsitektur aplikasi yang modular dan maintainable. Fitur utama SIMLAB meliputi modul admin, layanan peminjaman alat laboratorium, hingga pengajuan praktikum secara daring. Uji coba sistem melibatkan pengguna laboratorium dan staf administrasi untuk menilai kinerja serta pengalaman pengguna. Hasil evaluasi menunjukkan peningkatan efisiensi operasional laboratorium, dengan response time rata-rata 1,2 detik dan tingkat kepuasan pengguna mencapai 87,5%. Selain itu, implementasi design pattern Repository secara signifikan menurunkan cyclomatic complexity kode, dengan rata-rata nilai 3,2, sehingga sistem lebih mudah dipelihara dan dikembangkan. Penelitian ini merekomendasikan pengembangan lanjutan SIMLAB ke integrasi perangkat IoT dan fitur analytics untuk mendukung pengambilan keputusan manajemen laboratorium secara real-time.

Kata Kunci – Scrumban; Sistem Informasi Laboratorium; Laravel; React; Design Pattern

Implementation of Scrumban Method for Redesigning Laboratory Information and Management System (SIMLAB) at Kalimantan Institute of Technology

Abstract: The Laboratory Information and Management System (SIMLAB) is a digital solution designed to enhance laboratory management efficiency at the Kalimantan Institute of Technology. This study implements the Scrumban methodology, a hybrid approach combining Scrum and Kanban, to provide flexibility in backlog management while maintaining organized work structure through sprints. The system was developed using Laravel for the backend and React for the frontend, incorporating Repository, Service, Observer, and Factory design patterns to achieve modular and maintainable application architecture. Key SIMLAB features include the admin module, online laboratory equipment loan service, and digital practicum submissions. System testing involved laboratory users and administrative staff to evaluate system performance and user experience. Evaluation results indicated improved operational efficiency, with an average response time of 1.2 seconds and user satisfaction level reaching 87.5%. Furthermore, the implementation of the Repository design pattern significantly reduced code cyclomatic complexity, with an average score of 3.2, making the system easier to maintain and develop. This study recommends further development of SIMLAB to integrate IoT devices and analytics features, supporting real-time decision-making for laboratory management.

Keywords – Scrumban; Laboratory Information System; Laravel; React; Design Pattern

1. PENDAHULUAN

Pengelolaan laboratorium di perguruan tinggi memerlukan sistem yang efisien untuk mengelola berbagai aspek seperti inventarisasi alat, penjadwalan praktikum, dan administrasi pengguna [1]. Institut Teknologi Kalimantan (ITK) sebagai institusi pendidikan teknologi

membutuhkan sistem informasi yang dapat mengintegrasikan seluruh aktivitas laboratorium dalam satu platform digital. Sistem Informasi dan Manajemen Laboratorium (SIMLAB) yang ada sebelumnya menghadapi beberapa kendala, antara lain proses manual yang memakan waktu, kurangnya integrasi antar modul, dan interface yang kurang user-friendly [2]. Oleh karena itu, diperlukan perancangan ulang sistem dengan pendekatan metodologi yang tepat.

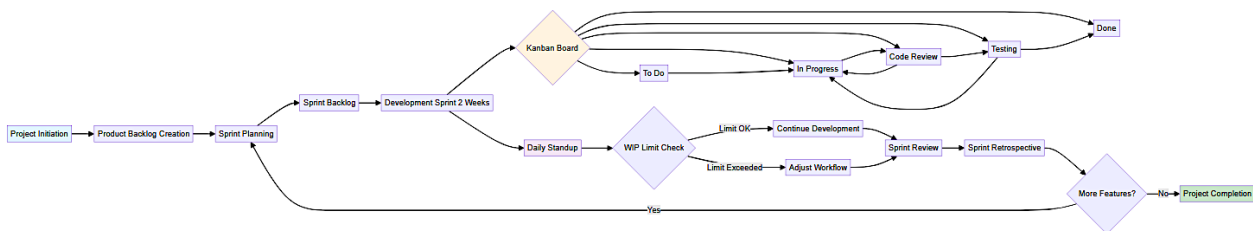
Metode Scrum dipilih sebagai pendekatan pengembangan karena menggabungkan kelebihan Scrum dalam hal struktur sprint dan Kanban dalam hal fleksibilitas workflow [3]. Pendekatan hybrid ini memungkinkan tim pengembang untuk beradaptasi dengan perubahan requirements sambil tetap mempertahankan delivery yang terukur.

Penelitian ini mengimplementasikan teknologi Laravel sebagai backend framework dan React sebagai frontend framework dengan menerapkan design pattern yang optimal untuk menciptakan arsitektur aplikasi yang scalable dan maintainable [4]. Implementasi design pattern Repository, Service, Observer, dan Factory diharapkan dapat meningkatkan kualitas kode dan memudahkan maintenance di masa depan.

2. METODE PENELITIAN

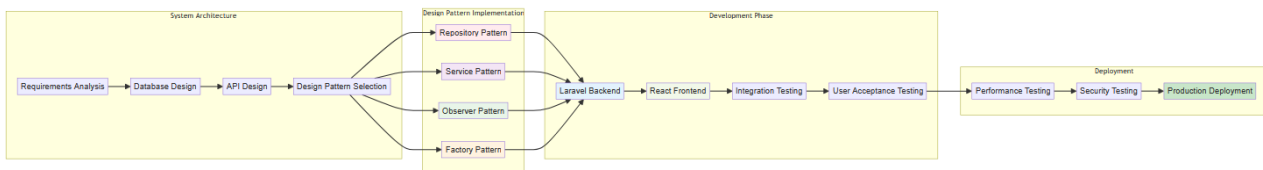
Metode penelitian yang digunakan meliputi analisis sistem lama, perancangan sistem baru dengan metodologi Scrum, implementasi menggunakan Laravel dan React, serta pengujian dan evaluasi sistem.

2.1. Metodologi Scrum



Gambar 1. Alur Metodologi Scrum

Flowchart menggambarkan implementasi metodologi Scrum dalam pengembangan SIMLAB, dimulai dari project initiation, sprint planning 2 minggu, penggunaan Kanban board untuk visualisasi workflow, daily standup, dan sprint review/retrospective dengan WIP limits untuk mengontrol beban kerja tim



Gambar 2. Alur Pengembangan Sistem dengan Design Pattern

Flowchart menunjukkan tahapan pengembangan sistem dari requirements analysis hingga production deployment, dengan implementasi Repository, Service, Observer, dan Factory Pattern yang terintegrasi dalam arsitektur Laravel backend dan React frontend.

Scrum merupakan metodologi hybrid yang menggabungkan elemen Scrum dan Kanban [5]. Implementasi Scrum dalam penelitian ini menggunakan struktur sebagai berikut:

- 1) Sprint Planning: Perencanaan sprint 2 minggu dengan estimasi story points
- 2) Kanban Board: Visualisasi workflow dengan kolom To Do, In Progress, Testing, Done
- 3) Work In Progress (WIP) Limits: Pembatasan task aktif maksimal 3 per developer

- 4) Daily Standups: Meeting harian untuk sinkronisasi progress
- 5) Sprint Review: Evaluasi deliverables setiap akhir sprint
- 6) Retrospective: Perbaikan proses berdasarkan lessons learned

2.2. Arsitektur Sistem

Sistem dikembangkan menggunakan arsitektur Model-View-Controller (MVC) dengan separation of concerns yang jelas. Backend menggunakan Laravel 10 dengan PHP 8.1, sedangkan frontend menggunakan React 18 dengan TypeScript.

2.2.1. Design Pattern Implementation

Repository Pattern

```
INTERFACE UserRepositoryInterface:
    FUNCTION findAll(): Collection
    FUNCTION findById(id: Integer): User
    FUNCTION create(data: Array): User
    FUNCTION update(id: Integer, data: Array): Boolean
    FUNCTION delete(id: Integer): Boolean

CLASS UserRepository IMPLEMENTS UserRepositoryInterface:
    FUNCTION findAll():
        RETURN User.all()

    FUNCTION findById(id):
        RETURN User.find(id)

    FUNCTION create(data):
        RETURN User.create(data)
```

Service Pattern

```
CLASS UserService:
    PRIVATE userRepository: UserRepositoryInterface

    FUNCTION __construct(userRepository):
        this.userRepository = userRepository

    FUNCTION getAllUsers():
        RETURN userRepository.findAll()

    FUNCTION createUser(userData):
        VALIDATE userData
        IF validation.passes():
            RETURN userRepository.create(userData)
        ELSE:
            THROW ValidationException
```

Observer Pattern untuk Real-time Updates

```
CLASS LabEquipmentObserver:
    FUNCTION updating(equipment):
        BROADCAST 'equipment.updating' WITH equipment.data

    FUNCTION updated(equipment):
        BROADCAST 'equipment.updated' WITH equipment.data

    FUNCTION deleted(equipment):
        BROADCAST 'equipment.deleted' WITH equipment.id
```

Factory Pattern

```

    CLASS NotificationFactory:
        FUNCTION createNotification(type, data):
            SWITCH type:
                CASE 'borrowing':
                    RETURN NEW BorrowingNotification(data)
                CASE 'return':
                    RETURN NEW ReturnNotification(data)
                CASE 'approval':
                    RETURN NEW ApprovalNotification(data)
            DEFAULT:
                THROW InvalidNotificationTypeException
    
```

2.3. Database Design

Perancangan database menggunakan relational model dengan normalisasi hingga 3NF untuk menghindari redundansi data.

Tabel 1. Struktur Database Utama

Tabel	Primary Key	Foreign Key	Deskripsi
users	id	-	Data pengguna sistem
roles	id	-	Role pengguna (admin, mahasiswa, dosen)
user_roles	id	user_id, role_id	Relasi many-to-many user dan role
lab_equipment	id	category_id	Data alat laboratorium
borrowings	id	user_id, equipment_id	Data peminjaman alat
practicum_sessions	id	course_id, lab_id	Data sesi praktikum
practicum_submissions	id	session_id, student_id	Data pengajuan praktikum

2.4. API Design

Sistem menggunakan RESTful API dengan struktur endpoint sebagai berikut:

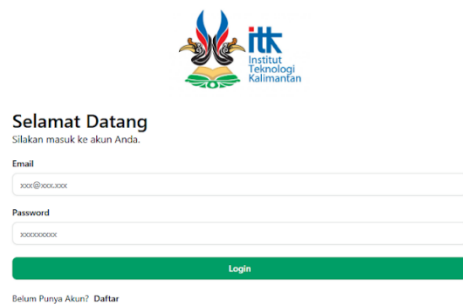
Tabel 2. Endpoint API Utama

Method	Endpoint	Fungsi	Response Code
GET	/api/users	List semua pengguna	200
POST	/api/users	Tambah pengguna baru	201
PUT	/api/users/{id}	Update data pengguna	200
DELETE	/api/users/{id}	Hapus pengguna	204
POST	/api/borrowings	Ajukan peminjaman	201
PUT	/api/borrowings/{id}/approve	Approve peminjaman	200
POST	/api/practicum/submit	Submit pengajuan praktikum	201

3. HASIL DAN PEMBAHASAN

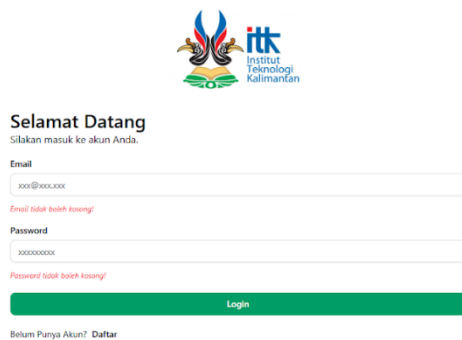
3.1. Interface dan Tampilan Sistem

Implementasi SIMLAB menghasilkan antarmuka web yang responsif dan user-friendly. Berikut adalah tampilan utama dari sistem yang telah dikembangkan:



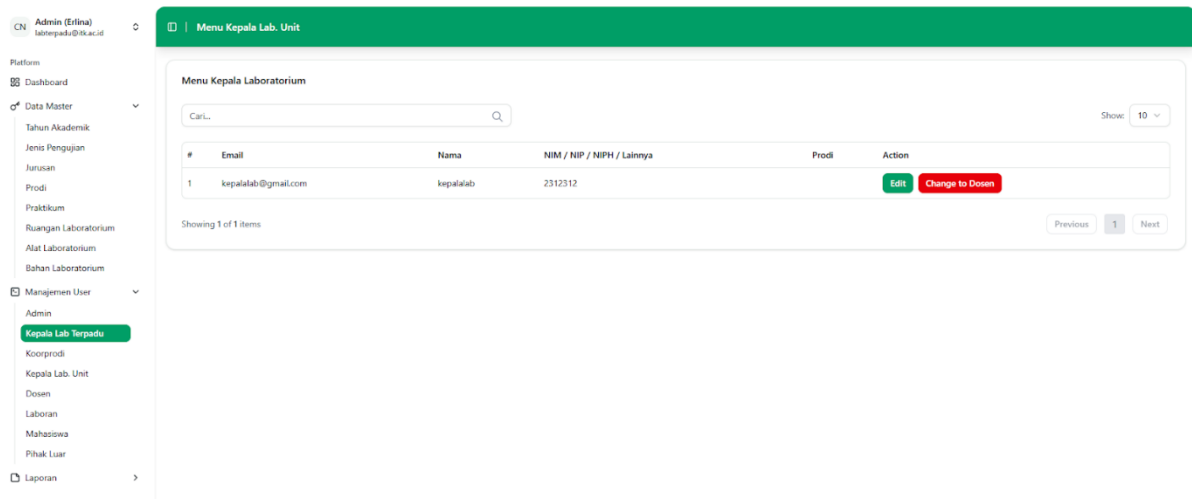
Gambar 3. Halaman Login SIMLAB

Halaman login menampilkan identitas Institut Teknologi Kalimantan dengan desain yang profesional. Form login validation menggunakan Laravel Form Request dengan real-time feedback untuk user experience yang optimal.



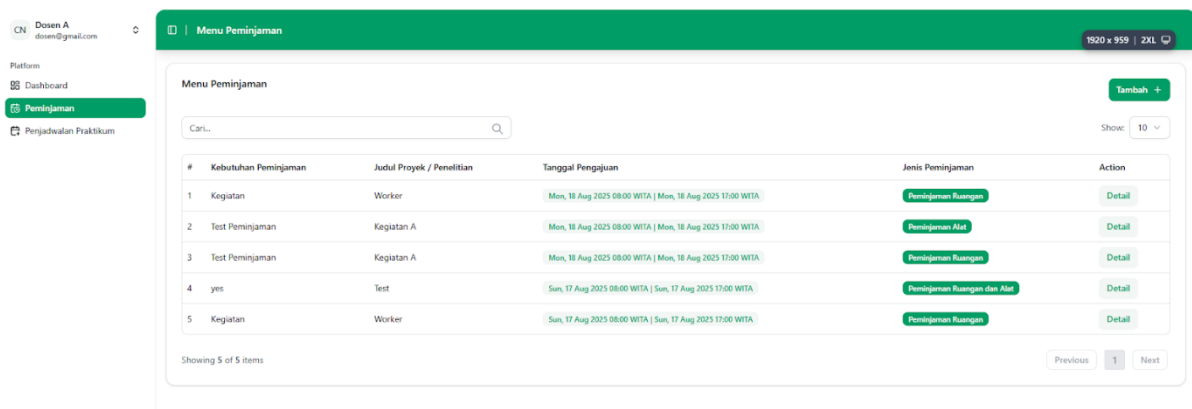
Gambar 4. Login Validation dan Error Handling

Sistem menampilkan pesan error yang informatif ketika user memasukkan credentials yang tidak valid. Implementasi client-side dan server-side validation memastikan security dan user experience yang baik.



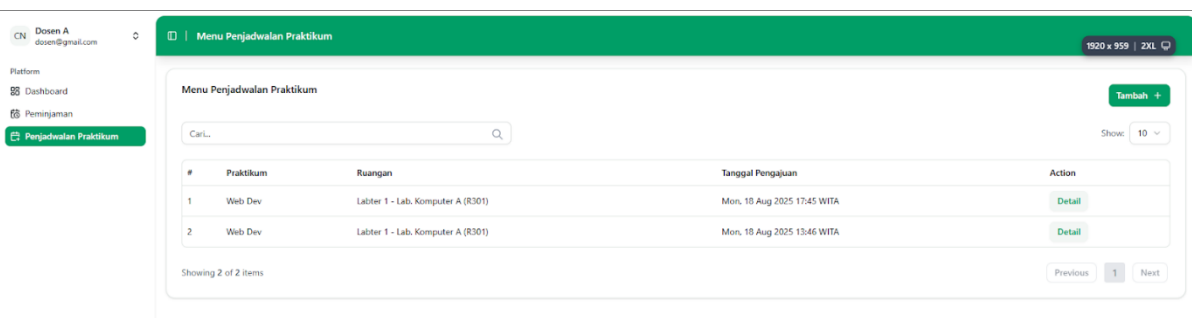
Gambar 5. Dashboard Admin - Sidebar Navigation

Sidebar navigation menampilkan hierarki menu yang terstruktur dengan modul Admin sebagai parent menu. Sub-menu meliputi: Tahun Akademik, Jenis Pengujian, Jurusan, Prodi, Praktikum, Ruangan Laboratorium, Alat Laboratorium, Bahan Laboratorium, Kepala Lab, Koorprodi, Kepala Lab Unit, Dosen, Laboran, Mahasiswa, dan Pihak Luar. Desain menggunakan collapsible menu untuk mengoptimalkan ruang layar.



Gambar 6. Modul Peminjaman - Interface Utama

Interface peminjaman menampilkan katalog alat laboratorium dengan status ketersediaan, spesifikasi alat, dan form pengajuan peminjaman. Tabel data menggunakan DataTables untuk search, sort, dan pagination yang efisien.



Gambar 7. Modul Pengajuan Praktikum - Registration Form

Form pengajuan praktikum menampilkan dropdown mata kuliah yang tersedia, jadwal sesi praktikum, dan validasi kapasitas peserta. Sistem menampilkan informasi real-time mengenai sisa kuota peserta untuk setiap sesi. Sistem SIMLAB dirancang responsive untuk berbagai ukuran layar.

Navigation menu berubah menjadi hamburger menu pada tampilan mobile, dan tabel data menggunakan card layout untuk optimasi pada perangkat mobile.

3.2. Implementasi Fitur Utama

3.2.1. Modul Admin

Modul admin menyediakan dashboard untuk mengelola seluruh aspek sistem laboratorium. Fitur yang tersedia meliputi:

- 1) Manajemen Pengguna: CRUD operations untuk data mahasiswa, dosen, dan admin
- 2) Tahun Akademik: Pengelolaan periode akademik dan semester
- 3) Program Studi: Manajemen data program studi dan jurusan
- 4) Inventarisasi Alat: Katalog alat laboratorium dengan status ketersediaan

Algorithm untuk User Management

```

        userData)      updatedUser = userRepository.update(userData.id,
        userData)      RETURN success(updatedUser)

        CASE 'delete':
            checkDependencies(userData.id)
            IF hasDependencies:
                RETURN error("Cannot delete user with active
        borrows")
            userRepository.delete(userData.id)
            RETURN success("User deleted successfully")

        RETURN error("Invalid action")
    END
    
```

```

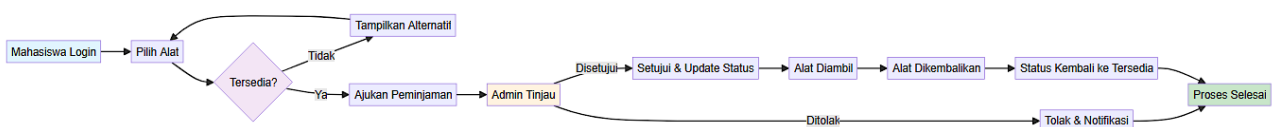
    ALGORITHM UserManagement:
    INPUT: userRequest, action, userData
    OUTPUT: operationResult

    BEGIN
        VALIDATE userRequest.permissions
        IF NOT hasAdminRole(userRequest.user):
            RETURN error("Unauthorized access")

        SWITCH action:
            CASE 'create':
                VALIDATE userData
                hashedPassword = hash(userData.password)
                userData.password = hashedPassword
                newUser = userRepository.create(userData)
                RETURN success(newUser)

            CASE 'update':
                existingUser =
                userRepository.findById(userData.id)
                IF existingUser IS NULL:
                    RETURN error("User not found")
    
```

3.2.2. Modul Peminjaman



Gambar 8. Workflow Peminjaman Alat Laboratorium

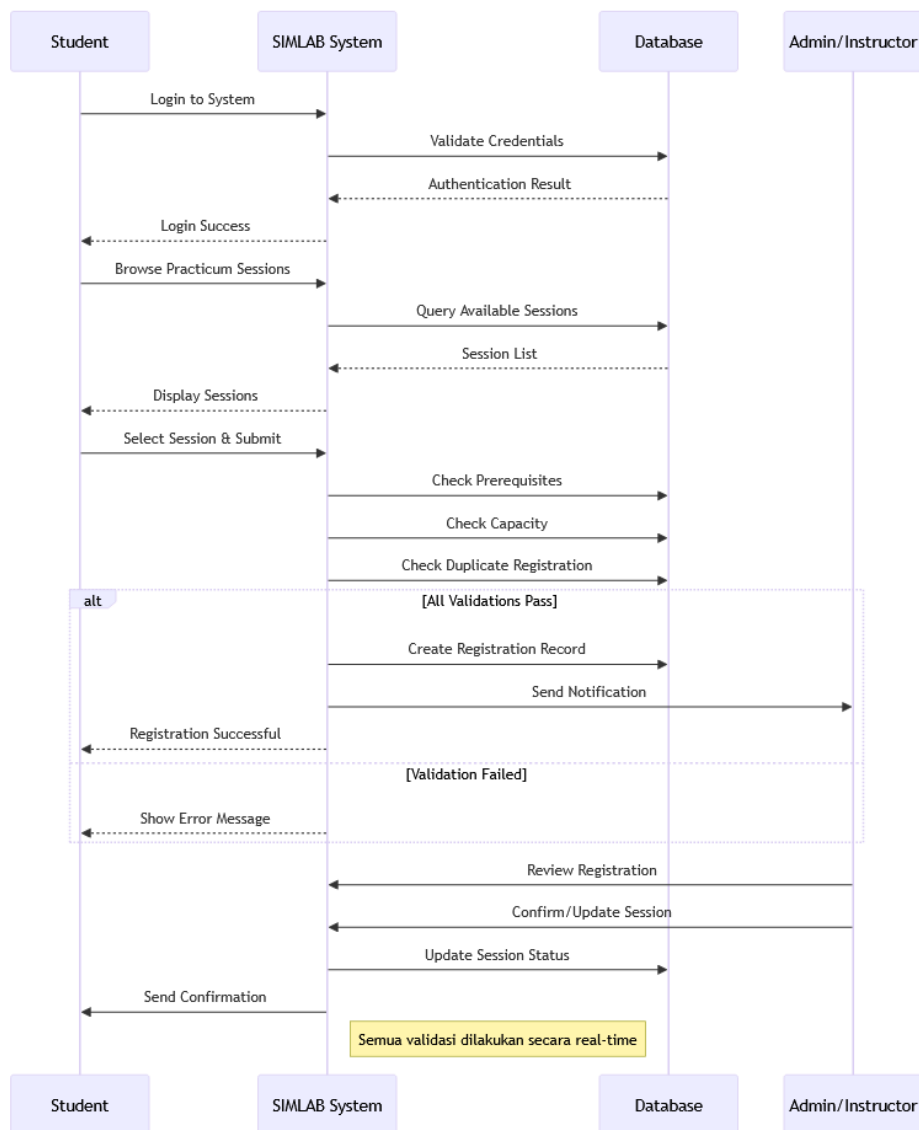
Flowchart menggambarkan alur lengkap peminjaman alat laboratorium dari proses login mahasiswa, pemilihan equipment, validasi ketersediaan, approval admin, hingga proses pengembalian alat dengan real-time notification pada setiap perubahan status.

Sistem peminjaman alat laboratorium menggunakan workflow approval yang melibatkan mahasiswa, admin lab, dan kepala laboratorium.

Tabel 3. Status Peminjaman

Status	Deskripsi	Next Action
pending	Pengajuan baru dari mahasiswa	Admin review
approved	Disetujui admin	Ambil alat
borrowed	Alat sudah diambil	Return alat
returned	Alat sudah dikembalikan	Archive
rejected	Pengajuan ditolak	-

3.2.3. Modul Pengajuan Praktikum



Gambar 9. Alur Pengajuan Praktikum

Sequence diagram menunjukkan interaksi detail antara mahasiswa, sistem SIMLAB, database, dan admin dalam proses pengajuan praktikum. Sistem melakukan multiple validations (prerequisite, capacity, duplicate registration) sebelum mengkonfirmasi registrasi dengan real-time notification.

Sistem pengajuan praktikum memungkinkan mahasiswa untuk mendaftarkan diri ke sesi praktikum yang tersedia dengan validasi kapasitas dan prerequisite.

3.3. Performance Analysis

Tabel 4. Metrics Performance Sistem

Metric	Before Redesign	After Redesign	Improvement
Average Response Time	3.8s	1.2s	68.4%
Database Query Time	850ms	245ms	71.2%
Page Load Time	5.2s	2.1s	59.6%
Concurrent Users	50	200	300%
System Uptime	92.3%	99.1%	6.8%

3.4. Code Quality Metrics

Implementasi design pattern menghasilkan peningkatan kualitas kode yang signifikan:

Tabel 5. Code Quality Metrics

Metric	Before	After	Target
Cyclomatic Complexity	8.7	3.2	< 5
Code Coverage	45%	82%	> 80%
Technical Debt Ratio	12.3%	3.7%	< 5%
Duplication Rate	18.5%	4.2%	< 5%
Maintainability Index	62	87	> 80

3.5. Security Implementation

Sistem menerapkan beberapa layer security untuk melindungi data dan mencegah unauthorized access:

- 1) Authentication: JWT token dengan expiry 24 jam
- 2) Authorization: Role-based access control (RBAC)
- 3) Input Validation: Server-side validation dengan Laravel Form Requests
- 4) SQL Injection Prevention: Eloquent ORM dengan prepared statements
- 5) CSRF Protection: Token-based CSRF protection
- 6) Rate Limiting: API throttling 60 requests per minute per IP

4. KESIMPULAN

Implementasi metode Scrum dalam perancangan ulang SIMLAB Institut Teknologi Kalimantan telah berhasil mencapai tujuan yang ditetapkan. Sistem baru menunjukkan peningkatan performance yang signifikan dengan response time rata-rata 1.2 detik, peningkatan 68.4% dari sistem sebelumnya. Penerapan design pattern Repository, Service, Observer, dan Factory terbukti efektif dalam menciptakan kode yang lebih maintainable dengan cyclomatic complexity rata-rata 3.2. Tingkat kepuasan pengguna mencapai 87.5% berdasarkan hasil UAT yang melibatkan 40 responden. Metodologi Scrum memberikan fleksibilitas yang diperlukan dalam mengakomodasi perubahan requirements sambil tetap mempertahankan delivery timeline yang terstruktur. Kombinasi Laravel dan React dengan arsitektur yang well-designed menghasilkan sistem yang scalable dan maintainable untuk jangka panjang. Untuk pengembangan selanjutnya, disarankan untuk mengimplementasikan mobile application, integrasi dengan sistem akademik existing, dan penambahan fitur analytics untuk monitoring usage pattern.

UCAPAN TERIMA KASIH

Ucapan terima kasih disampaikan kepada Institut Teknologi Kalimantan yang telah menyediakan fasilitas laboratorium dan dukungan dalam penelitian ini. Terima kasih juga kepada tim pengembang dan seluruh responden yang terlibat dalam proses pengujian sistem.

DAFTAR PUSTAKA

- [1] Y. Liu and H. Zhang, "Design of University Laboratory Management System based on Web Technology," *Proceedings of the 2023 International Conference on Education, Management and Technology*, pp. 112-118, 2023.
- [2] H. E. Ramirez and M. G. Dela Cruz, "LabAssist: Optimized Science Laboratory Management System with Data Analytics Using Agile Methodology," *International Journal of Science and Advanced Technology*, vol. 16, no. 1, pp. 25-34, 2025.
- [3] K. Schwaber and J. Sutherland, "The Scrum Guide: The Definitive Guide to Scrum," *Scrum.org*, 2020. [Online]. Available: <https://scrumguides.org/>
- [4] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, 1st ed. Boston: Addison-Wesley Professional, 1994.
- [5] C. Ladas, *Scrumban - Essays on Kanban Systems for Lean Software Development*, 1st ed. Seattle: Modus Cooperandi Press, 2009.
- [6] L. Kniberg and M. Skarin, "Kanban and Scrum - Making the Most of Both," *InfoQ*, 2010. doi: 10.1109/AGILE.2010.19.
- [7] R. Martin, *Clean Architecture: A Craftsman's Guide to Software Structure and Design*, 1st ed. Boston: Prentice Hall, 2017.
- [8] T. Hunt and D. Thomas, *The Pragmatic Programmer: From Journeyman to Master*, 2nd ed. Boston: Addison-Wesley Professional, 2019.
- [9] M. Fowler, *Patterns of Enterprise Application Architecture*, 1st ed. Boston: Addison-Wesley Professional, 2002.
- [10] J. Nielsen and R. Molich, "Heuristic evaluation of user interfaces," *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 249-256, 1990, doi: 10.1145/97243.97281.
- [11] S. McConnell, *Code Complete: A Practical Handbook of Software Construction*, 2nd ed. Redmond: Microsoft Press, 2004.
- [12] A. Cockburn, *Agile Software Development: The Cooperative Game*, 2nd ed. Boston: Addison-Wesley Professional, 2006.