



KERJA PRAKTIK – EF234603
Perancangan dan Pengembangan Sistem Manajemen dan Inventaris Properti pada Aplikasi Clusteria

PT. Aptikma Teknologi Indonesia

Jl. Wonorejo No 55. Kav. 6-8 Arjowinangun, Kota Malang, Jawa Timur

Periode: 15 Desember 2025 - 15 Maret 2026

Oleh:

Muhammad Rizqi Ar Rasyid 5025231062

Fadaukas Daffa Tajuddin 5025231149

Pembimbing Jurusan

Dr. Radityo Anggoro , S.Kom., M.Sc.

Pembimbing Lapangan

Amalia Wahyu Khoirunnikma

DEPARTEMEN TEKNIK INFORMATIKA

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya 2026

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

| | |
|---|-----------|
| DAFTAR ISI | 4 |
| DAFTAR GAMBAR | 8 |
| DAFTAR TABEL | 10 |
| LEMBAR PENGESAHAN | 12 |
| KATA PENGANTAR | 18 |
| BAB I | |
| PENDAHULUAN | 1 |
| 1.1. Latar Belakang | 1 |
| 1.2. Tujuan | 2 |
| 1.3. Manfaat | 2 |
| 1.4. Rumusan Masalah | 3 |
| 1.5. Lokasi dan Waktu Kerja Praktik | 3 |
| 1.6. Metodologi Kerja Praktik | 3 |
| 1.6.1. Perumusan Masalah | 3 |
| 1.6.2. Studi Literatur | 4 |
| 1.6.3. Analisis dan Perancangan Sistem | 4 |
| 1.6.4. Implementasi Sistem | 4 |
| 1.6.5. Pengujian dan Evaluasi | 4 |
| 1.6.6. Kesimpulan dan Saran | 5 |
| 1.7. Sistematika Laporan | 5 |
| 1.7.1. Bab I Pendahuluan | 5 |
| 1.7.2. Bab II Profil Perusahaan | 5 |
| 1.7.3. Bab III Tinjauan Pustaka | 5 |
| 1.7.4. Bab IV Analisis dan Perancangan Sistem | 5 |
| 1.7.5. Bab V Implementasi Sistem | 5 |
| 1.7.6. Bab VI Pengujian dan Evaluasi | 6 |

| | |
|--|-----------|
| 1.7.7. Bab VII Kesimpulan dan Saran | 6 |
| BAB II | |
| PROFIL PERUSAHAAN | 8 |
| 2.1. Profil PT. Aptikma Teknologi Indonesia | 8 |
| 2.2. Lokasi | 8 |
| BAB III | |
| TINJAUAN PUSTAKA | 10 |
| 3.1. Laravel | 10 |
| 3.2. MySQL | 10 |
| 3.3. Tailwind | 11 |
| BAB IV | |
| ANALISIS DAN PERANCANGAN INFRASTRUKTUR SISTEM | 13 |
| 4.1. Analisis Sistem | 13 |
| 4.1.1. Definisi Umum Aplikasi | 13 |
| 4.2. Perancangan Sistem | 14 |
| 4.2.1. Desain Arsitektur Sistem | 14 |
| 4.2.2. Perancangan Mekanisme Multi-Tenant | 15 |
| 4.2.3. Desain Basis Data | 16 |
| 4.2.4. Perancangan Keamanan Sistem | 19 |
| BAB V | |
| IMPLEMENTASI SISTEM | 22 |
| 5.1. Inisialisasi Lingkungan Pengembangan | 22 |
| 5.2. Implementasi Basis Data | 23 |
| 5.3. Implementasi Backend | 24 |
| 5.3.1. Implementasi Modul Gudang | 24 |
| 5.3.2. Implementasi Modul Penyesuaian Stok | 26 |
| 5.3.3. Implementasi Modul Purchase Request | 27 |
| 5.3.4. Implementasi Modul Purchase Order | 28 |

| | |
|---|-----------|
| 5.4.1. Integrasi API pada Frontend | 30 |
| 5.4.2. Implementasi Komponen Antarmuka Tabel Data | 31 |
| 5.4.3. Implementasi Form Dinamis Modul Penyesuaian Stok | 33 |
| 5.4.4. Implementasi Interaksi Aksi Purchase Order | 34 |
| BAB VI | |
| PENGUJIAN DAN EVALUASI | 36 |
| 6.1. Tujuan Pengujian | 36 |
| 6.2. Kriteria Pengujian | 37 |
| 6.3. Skenario Pengujian | 37 |
| 6.4. Evaluasi Pengujian | 38 |
| BAB VII | |
| KESIMPULAN DAN SARAN | 42 |
| 7.1. Kesimpulan | 42 |
| 7.2. Saran | 42 |
| DAFTAR PUSTAKA | 44 |
| BIODATA PENULIS I | 46 |
| BIODATA PENULIS II | 46 |

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

| | |
|--|----|
| Gambar 4.1. Desain Arsitektur Sistem Clusteria | 14 |
| Gambar 4.2. Alur Switching Database Perusahaan | 16 |
| Gambar 4.3. Tabel Master Database | 17 |
| Gambar 4.4. Tabel Company Database | 18 |

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 6.1. Hasil Evaluasi Pengujian

37

[Halaman ini sengaja dikosongkan]

**LEMBAR PENGESAHAN
KERJA PRAKTIK**

**Perancangan dan Pengembangan Sistem Manajemen
dan Inventaris Properti pada Aplikasi Clusteria**

Oleh:

Muhammad Rizqi Ar Rasyid
Fadaukas Daffa Tajuddin

5025231062
5025231149

Disetujui oleh Pembimbing Kerja Praktik:

1. Dr. Radityo Anggoro ,
S.Kom., M.Sc.
NIP. 198410162008121002


PT. APTIKMA TEKNOLOGI INDONESIA

(Pembimbing Departemen)

2. Amalia Wahyu
Khoirunnikma


PT. APTIKMA TEKNOLOGI INDONESIA

(Pembimbing Lapangan)

[Halaman ini sengaja dikosongkan]

Perancangan dan Pengembangan Sistem Manajemen dan Inventaris Properti pada Aplikasi Clusteria

Nama Mahasiswa : Muhammad Rizqi Ar Rasyid
NRP : 5025231062
Nama Mahasiswa : Fadaukas Daffa Tajuddin
NRP : 5025231149
Departemen : Teknik Informatika FTEIC-ITS
Pembimbing Departemen : Dr. Radityo Anggoro , S.Kom.,
M.Sc.
Pembimbing Lapangan : Amalia Wahyu Khoirunnikma

ABSTRAK

Clusteria merupakan aplikasi yang dirancang untuk mendukung pengelolaan properti pada lingkungan cluster secara terintegrasi. Dalam proses operasionalnya, pengelolaan data properti dan inventaris masih menghadapi berbagai kendala, seperti pencatatan yang belum terstruktur, kesulitan dalam pelacakan aset, serta kurangnya sistem yang mampu memberikan informasi secara real-time. Oleh karena itu, diperlukan suatu sistem manajemen dan inventaris properti yang terkomputerisasi guna meningkatkan efisiensi, akurasi, dan kemudahan dalam pengelolaan data.

Kerja praktik ini bertujuan untuk merancang dan mengembangkan sistem manajemen dan inventaris properti pada aplikasi Clusteria yang mampu mengelola data aset secara terpusat dan terintegrasi. Sistem ini dirancang untuk mendukung berbagai fitur utama seperti pencatatan data properti, pengelompokan inventaris, pemantauan status aset, serta pengelolaan informasi secara dinamis. Dengan adanya sistem ini,

diharapkan proses administrasi menjadi lebih efektif dan meminimalisir kesalahan dalam pengolahan data.

Sistem dikembangkan menggunakan arsitektur berbasis MVC (Model View Control) yang memisahkan logika aplikasi, tampilan, dan pengelolaan data sehingga memudahkan dalam pengembangan serta pemeliharaan sistem. Pengembangan backend dilakukan menggunakan framework Laravel, pengelolaan basis data menggunakan MySQL sebagai database relasional, serta antarmuka pengguna dibangun menggunakan Tailwind CSS untuk menghasilkan tampilan yang responsif dan modern. Kombinasi teknologi ini memungkinkan pengembangan sistem yang efisien, terstruktur, dan mudah untuk dikembangkan lebih lanjut.

Hasil dari kerja praktik ini adalah terbangunnya sistem manajemen dan inventaris properti yang terintegrasi dalam aplikasi Clusteria. Sistem ini mampu meningkatkan efisiensi dalam pengelolaan aset, mempermudah proses monitoring, serta mendukung pengambilan keputusan berbasis data. Selain itu, sistem yang dikembangkan memiliki struktur yang modular dan fleksibel sehingga mudah untuk dilakukan pemeliharaan dan pengembangan di masa mendatang.

Kata Kunci : Sistem Manajemen Properti, Inventaris, Laravel, MySQL.

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT atas penyertaan dan karunia-Nya sehingga penulis dapat menyelesaikan salah satu kewajiban penulis sebagai mahasiswa Departemen Teknik Informatika ITS yaitu Kerja Praktik yang berjudul: Perancangan dan Pengembangan Sistem Manajemen dan Inventaris Properti pada Aplikasi Clusteria.

Penulis menyadari bahwa masih banyak kekurangan baik dalam melaksanakan kerja praktik maupun penyusunan buku laporan kerja praktik ini. Namun penulis berharap buku laporan ini dapat menambah wawasan pembaca dan dapat menjadi sumber referensi.

Melalui buku laporan ini penulis juga ingin menyampaikan rasa terima kasih kepada orang-orang yang telah membantu menyusun laporan kerja praktik baik secara langsung maupun tidak langsung antara lain:

1. Kedua orang tua penulis.
2. Bapak Dr. Radityo Anggoro , S.Kom., M.Sc. selaku dosen pembimbing kerja praktik sekaligus koordinator kerja praktik.
3. Bapak Amalia Wahyu Khoirunnikma selaku pembimbing lapangan selama kerja praktik berlangsung.
4. Teman-teman penulis yang senantiasa memberikan semangat ketika penulis melaksanakan KP.

Surabaya, 16 Maret 2026

Muhammad Rizqi Ar Rasyid dan Fadaukas Daffa Tajuddin

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

1.1. Latar Belakang

Perkembangan teknologi informasi yang pesat telah mendorong berbagai sektor untuk mengadopsi sistem digital dalam mendukung kegiatan operasional, termasuk dalam pengelolaan properti dan inventaris. Pengelolaan aset yang masih dilakukan secara manual atau tidak terintegrasi seringkali menimbulkan berbagai permasalahan, seperti kesulitan dalam pencatatan data, kurangnya transparansi, serta keterbatasan dalam melakukan monitoring kondisi aset secara real-time. Hal ini dapat berdampak pada menurunnya efisiensi kerja serta meningkatnya risiko kesalahan dalam pengelolaan data.

Clusteria merupakan aplikasi yang dikembangkan untuk mendukung pengelolaan lingkungan cluster, khususnya dalam manajemen properti dan inventaris. Namun, dalam implementasinya masih ditemukan kebutuhan akan sistem yang mampu mengelola data properti dan inventaris secara terstruktur, terpusat, serta mudah diakses oleh pengguna. Sistem yang terintegrasi sangat diperlukan agar proses pencatatan, pemantauan, dan pengelolaan aset dapat dilakukan secara lebih efektif dan efisien.

Dalam pengembangan sistem modern, diperlukan perancangan yang tidak hanya berfokus pada fungsi utama, tetapi juga mencakup aspek arsitektur sistem, pengelolaan basis data, serta pengalaman pengguna. Penggunaan teknologi seperti Laravel sebagai framework backend, MySQL sebagai basis data relasional, serta Tailwind CSS untuk pengembangan frontend memungkinkan pembangunan sistem yang responsif, terstruktur, dan mudah dikembangkan.

Oleh karena itu, diperlukan perancangan dan pengembangan sistem manajemen dan inventaris properti pada aplikasi Clusteria yang mampu mengakomodasi kebutuhan pengelolaan aset secara dinamis. Sistem yang dirancang diharapkan dapat meningkatkan efisiensi operasional, mempermudah proses monitoring, serta mendukung pengambilan keputusan berbasis data secara akurat.

1.2. Tujuan

Tujuan dari perancangan dan pengembangan sistem manajemen dan inventaris properti pada aplikasi Clusteria adalah untuk membangun sistem yang terintegrasi, andal, dan terstruktur guna mendukung pengelolaan data properti dan inventaris secara efektif, termasuk pencatatan aset, pengelompokan data, pemantauan kondisi, serta penyajian informasi secara real-time, sehingga dapat meningkatkan efisiensi operasional dan kemudahan dalam pengelolaan sistem.

1.3. Manfaat

Manfaat dari perancangan dan pengembangan sistem manajemen dan inventaris properti pada aplikasi Clusteria adalah tersedianya sistem yang mampu membantu pengelolaan aset secara lebih terstruktur,

efisien, dan terintegrasi. Selain itu, sistem ini dapat meningkatkan akurasi data, mempermudah proses monitoring dan pelaporan, serta mendukung pengambilan keputusan yang lebih tepat. Dari sisi pengembangan, sistem ini juga memberikan kemudahan dalam proses pemeliharaan dan pengembangan fitur di masa mendatang.

1.4. Rumusan Masalah

Rumusan masalah dari kerja praktik ini adalah sebagai berikut:

1. Bagaimana merancang sistem manajemen dan inventaris properti yang terstruktur dan terintegrasi pada aplikasi Clusteria?
2. Bagaimana mengimplementasikan sistem yang mampu mengelola data properti dan inventaris secara efisien dan real-time?

1.5. Lokasi dan Waktu Kerja Praktik

Pengerjaan kerja praktik ini dilakukan secara *Hybrid*. Adapun kerja praktik dilaksanakan mulai tanggal 15 Desember 2025 hingga 15 Maret 2026.

1.6. Metodologi Kerja Praktik

Metodologi dalam penyusunan dan pelaksanaan kerja praktik meliputi beberapa tahapan berikut:

1.6.1. Perumusan Masalah

Tahap awal dilakukan melalui diskusi dengan pembimbing serta tim terkait untuk mengidentifikasi kebutuhan sistem. Pada tahap ini ditentukan fitur utama yang akan dikembangkan, seperti pengelolaan data properti, inventaris, serta monitoring aset.

1.6.2. Studi Literatur

Tahap ini dilakukan setelah mengetahui kebutuhan sistem dengan mempelajari berbagai referensi maupun dokumentasi terkait sistem manajemen inventaris, arsitektur aplikasi web, serta teknologi yang digunakan seperti Laravel dan MySQL. Selain itu, dipelajari juga konsep pengelolaan basis data dan pengembangan aplikasi berbasis web modern.

1.6.3. Analisis dan Perancangan Sistem

Pada tahap ini dilakukan analisis terhadap kebutuhan fungsional dan nonfungsional sistem. Berdasarkan hasil analisis, dirancang arsitektur sistem yang mencakup desain frontend dan backend, struktur API, alur komunikasi sistem, serta perancangan basis data. Perancangan difokuskan pada aspek skalabilitas, maintainability, konsistensi data, serta pengalaman pengguna.

1.6.4. Implementasi Sistem

Tahap implementasi dilakukan dengan mengembangkan sistem sesuai dengan desain yang telah dibuat. Backend dikembangkan menggunakan Laravel, database menggunakan MySQL, serta frontend menggunakan Tailwind CSS untuk menghasilkan tampilan yang responsif dan user-friendly.

1.6.5. Pengujian dan Evaluasi

Sistem yang telah diimplementasikan selanjutnya melalui tahap pengujian untuk memastikan seluruh fungsi berjalan sesuai kebutuhan. Pengujian dilakukan terhadap fitur aplikasi, komunikasi API, integrasi

frontend-backend, serta validasi alur sistem. Hasil pengujian digunakan sebagai dasar evaluasi guna mengidentifikasi potensi perbaikan dan optimasi sistem.

1.6.6. Kesimpulan dan Saran

Tahap akhir berupa penarikan kesimpulan berdasarkan hasil implementasi dan pengujian sistem. Selain itu, disusun saran yang dapat digunakan sebagai acuan untuk pengembangan dan peningkatan sistem di masa mendatang.

1.7. Sistematika Laporan

1.7.1. Bab I Pendahuluan

Bab ini berisi latar belakang, tujuan, manfaat, rumusan masalah, lokasi dan waktu kerja praktik, metodologi, dan sistematika laporan.

1.7.2. Bab II Profil Perusahaan

Bab ini berisi gambaran umum Telkom Indonesia mulai dari profil, lokasi perusahaan.

1.7.3. Bab III Tinjauan Pustaka

Bab ini berisi dasar teori dari teknologi yang digunakan dalam menyelesaikan proyek kerja praktik.

1.7.4. Bab IV Analisis dan Perancangan Sistem

Bab ini berisi mengenai tahap analisis sistem aplikasi dalam menyelesaikan proyek kerja praktik.

1.7.5. Bab V Implementasi Sistem

Bab ini berisi uraian tahap - tahap yang dilakukan untuk proses implementasi aplikasi.

1.7.6. Bab VI Pengujian dan Evaluasi

Bab ini berisi hasil uji coba dan evaluasi dari aplikasi yang telah dikembangkan selama pelaksanaan kerja praktik.

1.7.7. Bab VII Kesimpulan dan Saran

Bab ini berisi kesimpulan dan saran yang didapat dari proses pelaksanaan kerja praktik.

[Halaman ini sengaja dikosongkan]

BAB II

PROFIL PERUSAHAAN

2.1. Profil PT. Aptikma Teknologi Indonesia

PT Aptikma Teknologi Indonesia, didirikan atas dasar kecintaan terhadap dunia IT dan programming, bukan hanya sekedar berorientasi bisnis. Aptikma mulai beroperasi pada tahun 2015 di Kota Malang. Mulai tahun 2019, Aptikma memiliki kantor representative di Jakarta. Hal itu dikarenakan mayoritas client kami berasal dari Jakarta, sehingga diharapkan dapat mempermudah komunikasi dan koordinasi. Sesuai dengan perkembangan zaman, saat ini Aptikma berfokus pada Solusi AI dengan berbagai produk dan layanan yang kami berikan

2.2. Lokasi

Jl. Wonorejo No 55. Kav. 6-8 Arjowinangun,
Kota Malang, Jawa Timur.

[Halaman ini sengaja dikosongkan]

BAB III

TINJAUAN PUSTAKA

3.1. Laravel

Laravel merupakan framework aplikasi web berbasis PHP yang dirancang untuk mempermudah proses pengembangan aplikasi dengan sintaks yang elegan dan struktur yang terorganisir. Laravel mengadopsi pola arsitektur Model-View-Controller (MVC) yang memisahkan logika aplikasi, tampilan, dan pengelolaan data sehingga memudahkan dalam pengembangan serta pemeliharaan sistem.

Laravel menyediakan berbagai fitur yang mendukung pengembangan aplikasi modern, seperti routing, middleware, ORM (Object-Relational Mapping) melalui Eloquent, serta sistem templating Blade. Selain itu, Laravel juga memiliki dukungan terhadap keamanan aplikasi, seperti proteksi terhadap serangan SQL Injection, Cross-Site Scripting (XSS), dan Cross-Site Request Forgery (CSRF). Dengan fitur-fitur tersebut, Laravel menjadi salah satu framework yang banyak digunakan dalam pengembangan sistem berbasis web karena mampu meningkatkan produktivitas dan efisiensi pengembang.

3.2. MySQL

MySQL merupakan sistem manajemen basis data relasional (Relational Database Management System/RDBMS) yang digunakan untuk menyimpan dan mengelola data secara terstruktur. MySQL menggunakan bahasa SQL (Structured Query Language) sebagai standar dalam melakukan operasi pengolahan data, seperti

penyimpanan, pembaruan, penghapusan, dan pengambilan data.

Dalam pengembangan sistem informasi, MySQL banyak digunakan karena memiliki performa yang baik, mudah digunakan, serta mendukung integrasi dengan berbagai bahasa pemrograman, termasuk PHP yang digunakan dalam Laravel. MySQL juga mendukung konsep relasi antar tabel, sehingga sangat cocok digunakan untuk sistem manajemen inventaris yang membutuhkan keterkaitan data, seperti data properti, kategori aset, dan riwayat penggunaan.

3.3. Tailwind

Tailwind CSS merupakan framework CSS berbasis utility-first yang digunakan untuk membangun antarmuka pengguna secara cepat dan efisien. Berbeda dengan framework CSS tradisional, Tailwind CSS menyediakan kelas-kelas utilitas yang dapat langsung digunakan pada elemen HTML untuk mengatur tampilan, seperti warna, margin, padding, ukuran, dan layout.

Dengan pendekatan utility-first, pengembang tidak perlu menulis banyak kode CSS secara manual, sehingga proses pengembangan menjadi lebih cepat dan konsisten. Tailwind CSS juga mendukung pembuatan desain responsif dengan mudah melalui sistem breakpoint yang telah disediakan.

[Halaman ini sengaja dikosongkan]

BAB IV

ANALISIS DAN PERANCANGAN INFRASTRUKTUR SISTEM

4.1. Analisis Sistem

Pada bab ini dijelaskan tahapan analisis dan perancangan sistem manajemen dan inventaris properti pada aplikasi Clusteria. Analisis dilakukan untuk memahami kebutuhan sistem, baik dari sisi fungsional maupun nonfungsional, yang menjadi dasar dalam merancang aplikasi yang terintegrasi, efisien, dan mudah dikembangkan. Pembahasan pada bab ini meliputi definisi umum aplikasi, analisis kebutuhan sistem, serta perancangan arsitektur dan komponen sistem.

4.1.1. Definisi Umum Aplikasi

Secara umum, sistem manajemen dan inventaris properti pada aplikasi Clusteria merupakan sistem yang digunakan untuk mengelola data aset properti secara terpusat dan terstruktur. Sistem ini memungkinkan pengguna untuk melakukan pencatatan data properti, pengelompokan inventaris, pemantauan kondisi aset, serta pengelolaan informasi terkait secara real-time melalui antarmuka berbasis web.

Aplikasi ini terdiri dari beberapa komponen utama, yaitu frontend sebagai antarmuka pengguna, backend sebagai pengelola logika bisnis, serta basis data sebagai media penyimpanan informasi. Pengguna dapat mengakses sistem untuk melakukan pengelolaan data aset tanpa perlu melakukan proses manual, sehingga meningkatkan efisiensi dan akurasi dalam pengolahan data.

4.2. Perancangan Sistem

4.2.1. Desain Arsitektur Sistem

Clusteria menggunakan arsitektur aplikasi web berlapis (layered monolithic architecture) berbasis Laravel dengan pola Model-View-Controller (MVC). Arsitektur ini membagi sistem ke dalam beberapa lapisan utama, yaitu presentation layer, application layer, domain/data layer, dan data layer. Presentation layer terdiri dari Blade view, JavaScript, serta komponen frontend yang bertanggung jawab dalam menampilkan antarmuka pengguna. Application layer mencakup controller, middleware, validasi request, serta service pendukung yang mengatur alur logika aplikasi. Domain/data layer berisi model Eloquent, query builder, serta modul tambahan seperti ekspor data, sedangkan data layer mencakup basis data relasional serta media penyimpanan file.

Secara infrastruktur, sistem Clusteria memiliki dua konteks data utama, yaitu master database yang digunakan untuk menyimpan informasi platform atau perusahaan secara umum, serta company database yang digunakan untuk menyimpan data operasional masing-masing tenant atau perusahaan. Pemisahan ini memungkinkan pengelolaan data yang lebih terstruktur dan aman dalam lingkungan multi-tenant.



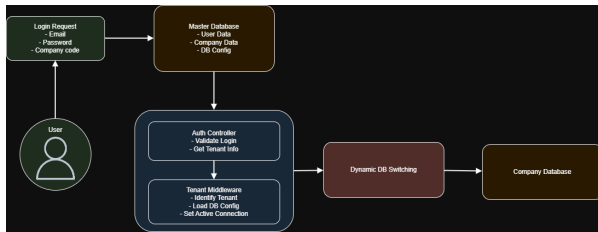
Gambar 4.1. Desain Arsitektur Sistem Clusteria

Dalam implementasinya, pengguna mengakses sistem melalui browser yang kemudian menampilkan antarmuka berbasis web. Setiap permintaan yang dilakukan pengguna akan diteruskan ke backend Laravel untuk diproses. Middleware berperan dalam memvalidasi sesi pengguna serta menentukan koneksi database perusahaan yang aktif. Selanjutnya, controller akan menjalankan logika bisnis dengan memanfaatkan model untuk berinteraksi dengan basis data. Data yang dihasilkan akan disimpan atau diambil dari database tenant sesuai dengan konteks perusahaan, kemudian hasilnya ditampilkan kembali ke frontend atau diekspor dalam bentuk laporan.

4.2.2. Perancangan Mekanisme Multi-Tenant

Sistem Clusteria menerapkan pendekatan database-per-company dalam mendukung arsitektur multi-tenant. Pada pendekatan ini, setiap perusahaan atau tenant memiliki basis data tersendiri yang terpisah dari tenant lainnya. Konfigurasi koneksi database perusahaan disimpan pada master database dan akan diakses ketika pengguna melakukan login sesuai dengan perusahaan yang dipilih.

Middleware dalam Laravel berperan penting dalam mekanisme ini, yaitu dengan melakukan switch koneksi database secara runtime. Setelah proses login berhasil, sistem akan membaca konfigurasi tenant dan mengatur koneksi database aktif sesuai dengan perusahaan pengguna. Dengan demikian, seluruh proses query berikutnya akan diarahkan ke database tenant yang sesuai tanpa perlu melakukan perubahan kode secara manual.



Gambar 4.2. Alur Switching Database Perusahaan

Pendekatan ini memberikan beberapa keuntungan, di antaranya adalah isolasi data antar perusahaan sehingga mengurangi risiko kebocoran data, peningkatan keamanan sistem, serta fleksibilitas dalam melakukan ekspansi jumlah tenant di masa mendatang. Selain itu, pendekatan ini juga memudahkan dalam pengelolaan data karena setiap tenant memiliki lingkungan data yang independen.

4.2.3. Desain Basis Data

Basis data pada sistem Clusteria menggunakan pendekatan relasional dengan MySQL atau MariaDB sebagai sistem manajemen basis data. Struktur basis data dirancang dengan membagi tabel ke dalam beberapa domain utama berdasarkan fungsi sistem, seperti data master, transaksi bisnis, stok, akuntansi, serta monitoring dan laporan.

Domain master data mencakup entitas seperti proyek, kavling, klien, supplier, material, gudang, serta berbagai data referensi lainnya. Domain transaksi bisnis mencakup proses operasional seperti SPPR, SPK, pembangunan, payment request, serta UTJ. Domain stok digunakan untuk mengelola histori stok, penyesuaian stok, serta transfer stok antar gudang. Domain akuntansi mencakup pengelolaan ledger, grup akun, jurnal, item jurnal, serta proses persetujuan

jurnal. Sementara itu, domain monitoring dan laporan digunakan untuk menyajikan data progres, pembayaran, serta ringkasan operasional secara menyeluruh.

MySQL > Server > Database: master_db_clusteria

Database: master_db_clusteria

[Alter database](#) [Database schema](#) [Privileges](#)

Tables and views

Search data in tables (0)

| <input type="checkbox"/> | Table | Engine | Collation? | Data Length | Index Length | Data Free | Auto Increment | Rows | Comment |
|--------------------------|----------------------|--------|--------------------|-------------|--------------|-----------|----------------|------|---------|
| <input type="checkbox"/> | company_log_activity | InnoDB | utf8mb4_unicode_ci | 16,384 | 16,384 | 0 | | 20 | ~ 13 |
| <input type="checkbox"/> | m_company | InnoDB | utf8mb4_0900_ai_ci | 16,384 | 0 | 0 | | 16 | ~ 12 |
| <input type="checkbox"/> | m_panduan_pdf | InnoDB | utf8mb4_0900_ai_ci | 65,536 | 0 | 0 | | 254 | ~ 252 |
| <input type="checkbox"/> | m_panduan_video | InnoDB | utf8mb4_0900_ai_ci | 16,384 | 0 | 0 | | 19 | ~ 18 |
| <input type="checkbox"/> | m_setting_kode | InnoDB | utf8mb4_0900_ai_ci | 16,384 | 0 | 0 | | 43 | ~ 42 |
| <input type="checkbox"/> | m_user_mdb | InnoDB | utf8mb4_0900_ai_ci | 16,384 | 0 | 0 | | 1 | 0 |
| <input type="checkbox"/> | ref_setting_tipe | InnoDB | utf8mb4_0900_ai_ci | 16,384 | 0 | 0 | | 7 | ~ 7 |
| <input type="checkbox"/> | ref_tipe_company | InnoDB | utf8mb4_0900_ai_ci | 16,384 | 0 | 0 | | 4 | ~ 3 |
| # | in total | InnoDB | utf8mb4_0900_ai_ci | 180,224 | 16,384 | 0 | | | |

Gambar 4.3. Tabel Master Database

Secara umum, desain basis data menggunakan primary key berupa nilai numerik pada setiap entitas utama untuk memastikan identifikasi data yang unik. Sistem juga mendukung mekanisme workflow melalui penggunaan field status dan tabel detail status, sehingga proses bisnis dapat dipantau dengan jelas. Selain itu, sistem menyediakan fitur audit dasar melalui pencatatan data created dan edited untuk setiap perubahan data. Dukungan terhadap penyimpanan dokumen atau foto juga disediakan sebagai bagian dari kebutuhan proses bisnis yang memerlukan lampiran data.

4.2.4. Perancangan Keamanan Sistem

Perancangan keamanan sistem pada Clusteria dilakukan untuk memastikan bahwa data dan proses bisnis terlindungi dengan baik dari akses yang tidak sah. Sistem menerapkan mekanisme autentikasi pengguna melalui proses login berbasis sesi, sehingga hanya pengguna yang terdaftar yang dapat mengakses aplikasi.

Selain itu, sistem juga menerapkan otorisasi berbasis peran (role-based access control) untuk mengatur hak akses pengguna terhadap menu dan aksi tertentu. Middleware digunakan untuk membatasi akses terhadap route tertentu sesuai dengan peran pengguna yang sedang aktif. Dengan demikian, setiap pengguna hanya dapat mengakses fitur yang sesuai dengan kewenangannya.

Keamanan sistem juga diperkuat dengan penerapan isolasi data antar tenant melalui mekanisme pemilihan koneksi database berdasarkan perusahaan aktif. Hal ini memastikan

bahwa data dari satu perusahaan tidak dapat diakses oleh perusahaan lain. Selain itu, sistem juga menerapkan kontrol pada proses-proses kritikal, seperti persetujuan jurnal dan pembayaran, melalui mekanisme approval untuk memastikan validitas dan keamanan transaksi yang dilakukan.

[Halaman ini sengaja dikosongkan]

BAB V

IMPLEMENTASI SISTEM

Bab ini menjelaskan implementasi sistem pada aplikasi yang dikembangkan selama kegiatan magang. Fokus pembahasan meliputi implementasi lingkungan pengembangan, implementasi basis data, implementasi backend, serta implementasi antarmuka untuk modul Stok dan modul Purchase.

5.1. Inisialisasi Lingkungan Pengembangan

Lingkungan pengembangan disiapkan untuk mendukung proses pengembangan fitur, pengujian, dan debugging secara konsisten. Komponen utama yang digunakan dalam pengembangan sistem ini meliputi bahasa pemrograman PHP, framework Laravel, basis data relasional MySQL, serta Node.js untuk pengelolaan aset frontend menggunakan Vite dan Tailwind CSS. Selain itu, digunakan juga Composer sebagai dependency manager untuk PHP dan NPM untuk pengelolaan dependency frontend.

Tahap inisialisasi dilakukan dengan melakukan konfigurasi awal proyek Laravel, instalasi dependency, serta pengaturan koneksi basis data. Konfigurasi variabel lingkungan dikelola melalui berkas `.env`, yang mencakup parameter koneksi database, konfigurasi aplikasi, serta pengaturan lainnya yang diperlukan selama proses pengembangan.

```
composer install
cp .env.example .env
php artisan key:generate
php artisan migrate
php artisan serve
```

Kode Sumber 5.1 Inisialisasi Project

5.2. Implementasi Basis Data

Basis data diimplementasikan menggunakan pendekatan relasional untuk menjaga integritas data antar modul. Pada cakupan pembahasan ini, entitas utama yang digunakan meliputi master gudang, master stok material per gudang, header dan detail penyesuaian stok, header dan detail purchase request, header dan detail purchase order, serta riwayat pergerakan stok.

Struktur data dirancang untuk mendukung berbagai kebutuhan operasional, seperti pelacakan stok berdasarkan material dan gudang, pengelolaan alur persetujuan pengadaan, keterkaitan dokumen dari permintaan pembelian hingga proses penerimaan barang, serta pencatatan audit trail terhadap perubahan kuantitas stok. Dengan desain ini, sistem mampu menjaga konsistensi data sekaligus mendukung proses bisnis yang berjalan secara terintegrasi.

```
<?php
class Warehouse extends Model
{
    protected $table = 'warehouses';
    protected $primaryKey = 'warehouse_id';
    public $timestamps = false;

    protected $fillable = [
        'warehouse_code',
        'warehouse_name',
        'address',
        'is_active'
    ];

    public function scopeActive($query)
```

```

    {
        return $query->where('is_active', 1);
    }
}

```

Kode Sumber 5.2. Model Entitas Gudang

```

<?php
class MaterialStock extends Model
{
    protected $table = 'material_stocks';
    protected $primaryKey = 'stock_id';
    public $timestamps = false;

    protected $fillable = [
        'material_id',
        'warehouse_id',
        'qty_base',
        'last_update'
    ];
}

```

Kode Sumber 5.3. Model Entitas Stok

5.3. Implementasi Backend

Implementasi backend dilakukan menggunakan pola arsitektur MVC (Model-View-Controller) untuk menjaga pemisahan tanggung jawab antar komponen. Pada pola ini, controller bertugas menangani request serta alur proses aplikasi, model bertanggung jawab dalam pengelolaan data dan relasi antar entitas, sedangkan view digunakan untuk menampilkan data kepada pengguna.

5.3.1. Implementasi Modul Gudang

Modul gudang digunakan untuk mendukung pengelolaan data gudang, yang mencakup proses penambahan, pembaruan, penampilan, serta penonaktifan

data gudang. Pada saat gudang baru dibuat, sistem secara otomatis menyiapkan data stok awal untuk material yang aktif, sehingga gudang tersebut dapat langsung digunakan dalam transaksi operasional berikutnya.

```
<?php
public function store(Request $request)
{
    $request->validate([
        'warehouse_code' => 'required',
        'warehouse_name' => 'required'
    ]);

    $warehouse = Warehouse::create([
        'warehouse_code' => $request->warehouse_code,
        'warehouse_name' => $request->warehouse_name,
        'address' => $request->address,
        'is_active' => 1
    ]);

    $materials = Material::active()->get();

    foreach ($materials as $material) {
        MaterialStock::create([
            'warehouse_id' => $warehouse->warehouse_id,
            'material_id' => $material->material_id,
            'qty_base' => 0,
            'last_update' => now()
        ]);
    }

    return redirect()->route('warehouse.index');
}
```

Kode Sumber 5.4. Implementasi Modul Gudang

5.3.2. Implementasi Modul Penyesuaian Stok

Modul penyesuaian stok digunakan untuk mencatat koreksi stok, baik penambahan maupun pengurangan jumlah barang. Implementasi modul ini menggunakan transaksi basis data untuk memastikan bahwa perubahan pada data header, detail, histori, serta saldo stok dilakukan secara konsisten dan terintegrasi.

```
<?php
DB::beginTransaction();

try {
    $adjustment = StockAdjustment::create([
        'adjustment_code' => generateCode('ADJ'),
        'warehouse_id' => $request->warehouse_id,
        'adjustment_type' => $request->adjustment_type, // IN /
OUT
        'adjustment_date' => $request->adjustment_date,
        'notes' => $request->notes
    ]);

    foreach ($request->items as $item) {
        $qtyBase = convertToBaseUnit($item['qty'],
$item['unit_id'], $item['material_id']);

        StockAdjustmentDetail::create([
            'adjustment_id' => $adjustment->adjustment_id,
            'material_id' => $item['material_id'],
            'unit_id' => $item['unit_id'],
            'qty_base' => $qtyBase
        ]);

        updateStockBalance($request->warehouse_id,
$item['material_id'], $qtyBase, $request->adjustment_type);
        writeStockHistory($request->warehouse_id,
$item['material_id'], $qtyBase, $request->adjustment_type);
    }

    DB::commit();
}
```

```

        return back()->with('success', 'Penyesuaian stok berhasil
disimpan.');
```

```

    } catch (\Throwable $e) {
        DB::rollBack();
        return back()->with('error', 'Penyesuaian stok gagal
diproses.');
```

```

    }

```

Kode Sumber 5.5. Implementasi Modul Penyesuaian Stok

5.3.3. Implementasi Modul Purchase Request

Purchase Request digunakan untuk mencatat kebutuhan pembelian barang atau material dari unit operasional. Setiap purchase request memiliki nomor dokumen yang unik, data penerima, serta detail item yang dibutuhkan. Modul ini berperan sebagai tahap awal dalam proses pengadaan barang.

```

<?php
public function createPurchaseRequest(Request $request)
{
    $pr = PurchaseRequest::create([
        'pr_number' => generateDocumentNumber('PR'),
        'request_date' => now(),
        'requestor_id' => auth()->id(),
        'status_code' => 'NEW'
    ]);

    foreach ($request->items as $item) {
        PurchaseRequestDetail::create([
            'pr_id' => $pr->pr_id,
            'material_id' => $item['material_id'],
            'qty' => $item['qty'],
            'unit_id' => $item['unit_id'],
            'supplier_id' => $item['supplier_id'] ?? null,
            'remarks' => $item['remarks'] ?? null
        ]);
    }
}

```

```

    }

    return redirect()->route('purchase-request.index');
}

```

Kode Sumber 5.6. Implementasi Purchase Request

5.3.4. Implementasi Modul Purchase Order

Purchase Order dibentuk berdasarkan purchase request yang telah diproses. Sistem akan menyusun data pemesanan berdasarkan pemasok, menghitung nilai transaksi, serta menyiapkan dokumen lanjutan yang diperlukan dalam proses penerimaan barang.

```

<?php
public function generatePurchaseOrder(int $prId)
{
    $pr = PurchaseRequest::with('details')->findOrFail($prId);
    $groupedBySupplier = $pr->details->groupBy('supplier_id');

    foreach ($groupedBySupplier as $supplierId => $items) {
        if (!$supplierId) {
            continue;
        }

        $subtotal = 0;
        foreach ($items as $item) {
            $subtotal += ($item->qty *
getPrice($item->material_id, $item->unit_id));
        }

        $po = PurchaseOrder::create([
            'po_number' => generateDocumentNumber('PO'),
            'pr_id' => $pr->pr_id,
            'supplier_id' => $supplierId,
            'subtotal' => $subtotal,
            'total' => $subtotal,

```

```

        'status_code' => 'NEW'
    ]);

    foreach ($items as $item) {
        PurchaseOrderDetail::create([
            'po_id' => $po->po_id,
            'material_id' => $item->material_id,
            'qty' => $item->qty,
            'unit_id' => $item->unit_id,
            'unit_price' => getPrice($item->material_id,
$item->unit_id),
            'line_total' => $item->qty *
getPrice($item->material_id, $item->unit_id)
        ]);
    }
}

return back()->with('success', 'Purchase Order berhasil
dibuat.');
```

Kode Sumber 5.7. Implementasi Purchase Order

5.4. Implementasi Frontend

Pada tahap implementasi frontend, sistem dirancang untuk mendukung aktivitas operasional harian pengguna melalui antarmuka berbasis web. Frontend berfungsi sebagai media interaksi antara pengguna dan backend, khususnya dalam proses input data, validasi, pengolahan transaksi, serta monitoring status proses bisnis pada modul Stok dan Purchase.

Implementasi frontend pada sistem ini menggunakan pendekatan server-rendered view yang dipadukan dengan komponen JavaScript untuk mendukung interaksi dinamis. Pendekatan ini memungkinkan proses seperti pengisian data secara bertahap, pemanggilan data secara asinkron, serta pembaruan tabel tanpa perlu melakukan reload halaman

secara penuh, sehingga meningkatkan efisiensi dan pengalaman pengguna.

5.4.1. Integrasi API pada Frontend

Integrasi antara frontend dan backend dilakukan melalui pemanggilan endpoint HTTP untuk mengambil maupun mengirimkan data. Mekanisme ini digunakan dalam berbagai proses, seperti pengambilan daftar data gudang dan stok, validasi stok saat penyesuaian, pengambilan detail Purchase Request dan Purchase Order, serta pengiriman data form transaksi.

Sebagai contoh, pada proses validasi stok material sebelum penyimpanan penyesuaian, frontend akan mengirimkan data berupa material, gudang, dan jumlah penyesuaian ke endpoint backend. Backend kemudian memproses data tersebut dan mengembalikan hasil validasi untuk menentukan apakah transaksi dapat dilanjutkan. Hasil respons tersebut selanjutnya ditampilkan langsung pada antarmuka sebagai umpan balik kepada pengguna.

```
async function validateStock(payload) {
  try {
    const response = await
fetch("/stok/penyesuaian/check-stock", {
  method: "POST",
  headers: {
    "Content-Type": "application/json",
    "X-CSRF-TOKEN": document
      .querySelector('meta[name="csrf-token"]')
      .getAttribute("content"),
  },
  body: JSON.stringify(payload),
});
```

```

const result = await response.json();

if (!result.success) {
  showAlert("error", result.message || "Validasi stok
gagal.");
  return null;
}

return result;
} catch (error) {
  console.error("[validateStock] Error:", error);
  showAlert("error", "Terjadi kesalahan saat memeriksa
stok.");
  return null;
}
}
}

```

Kode Sumber 5.8. Implementasi Pemanggilan Endpoint pada Frontend

5.4.2. Implementasi Komponen Antarmuka Tabel Data

Komponen tabel digunakan untuk menampilkan data transaksi dalam jumlah besar dengan tetap mempertahankan performa dan kemudahan penggunaan. Tabel dilengkapi dengan fitur pencarian, pengurutan, serta filter data yang membantu pengguna dalam memantau dan mengelola data operasional secara lebih efektif.

Sebagai contoh, pada halaman Purchase Request, tabel menampilkan daftar PR menggunakan pendekatan server-side processing sehingga tetap optimal meskipun jumlah data meningkat. Selain itu, tabel juga menyediakan kolom aksi yang berisi tombol seperti detail, edit, approve, atau delete yang ditampilkan sesuai dengan hak akses pengguna.

```

<div class="card">
  <div class="card-header">
    <h5>Daftar Purchase Request</h5>
  </div>
  <div class="card-body">
    <table id="table-pr" class="table table-bordered
table-striped">
      <thead>
        <tr>
          <th>No PR</th>
          <th>Tanggal</th>
          <th>Pemohon</th>
          <th>Status</th>
          <th>Aksi</th>
        </tr>
      </thead>
      <tbody></tbody>
    </table>
  </div>
</div>

$(function () {
  $("#table-pr").DataTable({
    processing: true,
    serverSide: true,
    ajax: "/purchase/purchase_requests",
    columns: [
      { data: "pr_number", name: "pr_number" },
      { data: "request_date", name: "request_date" },
      { data: "requestor_name", name: "requestor_name" },
      { data: "status_label", name: "status_label" },
      { data: "action", name: "action", orderable: false,
searchable: false },
    ],
  });
});

```

Kode Sumber 5.9. Implementasi Komponen Komponen Tabel Frontend

5.4.3. Implementasi Form Dinamis Modul Penyesuaian Stok

Pada modul Penyesuaian Stok, frontend dirancang untuk mendukung input multi-item dalam satu transaksi. Pengguna dapat menambahkan beberapa baris material dalam satu dokumen, memilih satuan, memasukkan jumlah, serta melakukan validasi stok sebelum data disimpan.

Pendekatan form dinamis ini memungkinkan fleksibilitas dalam proses input, terutama pada kondisi lapangan di mana satu transaksi seringkali melibatkan lebih dari satu jenis material. Dengan demikian, pengguna dapat mengelola data dengan lebih efisien tanpa perlu membuat dokumen terpisah untuk setiap item.

```
function addMaterialRow() {
  const row = `
    <tr>
      <td><select name="items[][material_id]"
class="form-control material-select"></select></td>
      <td><select name="items[][unit_id]" class="form-control
unit-select"></select></td>
      <td><input type="number" name="items[][qty]"
class="form-control qty-input" min="0.01" step="0.01"></td>
      <td><button type="button" class="btn btn-danger btn-sm
btn-remove-row">Hapus</button></td>
    </tr>
  `;
  document.querySelector("#materials-table
tbody").insertAdjacentHTML("beforeend", row);
}
```

Kode Sumber 5.10. Implementasi Form Dinamis Penyesuaian Stok

5.4.4. Implementasi Interaksi Aksi Purchase Order

Pada modul Purchase Order, frontend menyediakan aksi yang bersifat kontekstual berdasarkan status dokumen. Setiap status memiliki aksi yang berbeda, seperti pada status baru ditampilkan aksi invoicing, pada status tervalidasi ditampilkan aksi penerimaan barang, dan pada status selesai hanya tersedia opsi untuk melihat detail serta mencetak dokumen.

Pengendalian aksi ini bertujuan untuk menjaga alur proses bisnis agar tetap terstruktur dan sesuai dengan tahapan yang telah ditentukan. Dengan demikian, pengguna hanya dapat menjalankan aksi yang relevan dengan kondisi dokumen saat ini, sehingga meminimalisir kesalahan dalam proses operasional.

```
function renderActionButtons(row) {
  let actions = `
```

Kode Sumber 5.11. Implementasi Interaksi Aksi Purchase Order

[Halaman ini sengaja dikosongkan]

BAB VI

PENGUJIAN DAN EVALUASI

Bab ini menjelaskan proses pengujian terhadap sistem Clusteria dengan fokus pada modul Stok serta modul Purchase. Pengujian dilakukan untuk memastikan bahwa sistem yang telah diimplementasikan dapat berjalan sesuai dengan rancangan yang telah ditetapkan pada tahap analisis dan perancangan. Selain itu, pengujian juga bertujuan untuk memverifikasi bahwa sistem mampu menjalankan fungsi-fungsi inti dengan baik serta dapat berinteraksi dengan basis data dan komponen sistem lainnya secara konsisten.

6.1. Tujuan Pengujian

Pengujian pada sistem Clusteria dilakukan untuk memastikan bahwa seluruh modul dapat berfungsi dengan baik sesuai dengan kebutuhan sistem. Secara khusus, pengujian bertujuan untuk memastikan bahwa modul Stok mampu mengelola data gudang dan perhitungan stok dengan akurat, modul Penyesuaian Stok dapat mencatat mutasi stok masuk dan keluar serta memperbarui saldo stok secara real-time, modul Purchase Request mampu mengelola permintaan pembelian dan proses persetujuan secara terstruktur, serta modul Purchase Order dapat menghasilkan dokumen pemesanan berdasarkan PR yang telah disetujui dengan nilai yang sesuai.

Selain itu, pengujian juga bertujuan untuk memastikan bahwa integrasi antar modul, khususnya antara modul Purchase dan Stok, dapat berjalan tanpa konflik data. Sistem juga diuji untuk memastikan bahwa basis data relasional mampu menyimpan dan mengambil data secara konsisten.

6.2. Kriteria Pengujian

Penilaian terhadap keberhasilan sistem dilakukan berdasarkan beberapa kriteria utama, yaitu kemampuan sistem dalam menampilkan antarmuka aplikasi secara responsif, melakukan validasi input pada form entri data, serta menyimpan data transaksi ke dalam basis data dengan benar. Selain itu, sistem juga harus mampu melakukan perhitungan stok dasar dengan konversi satuan yang tepat, mengelola workflow approval pada Purchase Request, serta menghasilkan Purchase Order secara otomatis berdasarkan PR yang telah disetujui.

Kriteria lainnya mencakup kemampuan sistem dalam melacak perubahan stok melalui histori mutasi serta menangani transaksi yang berjalan secara bersamaan (concurrent) tanpa menyebabkan kerusakan data (data corruption).

6.3. Skenario Pengujian

Skenario pengujian dilakukan dengan mensimulasikan aktivitas pengguna dalam mengoperasikan fitur pada modul Stok dan Purchase sesuai dengan alur proses bisnis yang telah dirancang.

Pada modul gudang, pengujian dilakukan melalui pembuatan dan perubahan data gudang. Pengguna dapat menambahkan data gudang baru dengan mengisi informasi yang diperlukan, kemudian sistem akan menyimpan data serta secara otomatis menyiapkan stok awal untuk material aktif. Selain itu, pengguna juga dapat melakukan perubahan data gudang, dan sistem akan memperbarui data sesuai input yang diberikan.

Pada modul penyesuaian stok, pengujian dilakukan untuk dua kondisi, yaitu penambahan dan pengurangan stok. Pada proses penambahan, sistem akan menyimpan data transaksi, memperbarui saldo stok, serta

mencatat histori mutasi. Sementara pada proses pengurangan, sistem akan melakukan validasi terhadap ketersediaan stok. Jika jumlah yang diminta melebihi stok yang tersedia, sistem akan menolak transaksi dan menampilkan pesan kesalahan hingga data diperbaiki.

Pada modul Purchase Request, pengujian dilakukan melalui pembuatan dokumen permintaan pembelian serta proses persetujuan. Sistem akan menghasilkan nomor dokumen secara otomatis dan menetapkan status awal. Setelah dilakukan persetujuan oleh pengguna yang berwenang, sistem akan memperbarui status dan secara otomatis menghasilkan Purchase Order berdasarkan data yang telah disetujui.

Pada modul Purchase Order, pengujian mencakup verifikasi pembuatan dokumen secara otomatis, proses validasi invoice, serta penerimaan barang. Sistem akan memastikan kesesuaian data antara Purchase Request dan Purchase Order, memproses validasi dokumen, serta memperbarui stok setelah proses penerimaan barang dilakukan.

6.4. Evaluasi Pengujian

Hasil pengujian dilakukan dengan mengamati perilaku sistem terhadap skenario yang telah dijalankan. Berdasarkan hasil evaluasi, sistem mampu memenuhi seluruh kriteria pengujian yang telah ditetapkan. Sistem dapat menampilkan antarmuka dengan baik, melakukan validasi input secara tepat, serta menyimpan data transaksi secara konsisten.

Selain itu, sistem juga mampu melakukan perhitungan stok dengan konversi satuan secara akurat, memvalidasi ketersediaan stok sebelum proses pengurangan, serta menghasilkan Purchase Request dan Purchase Order sesuai dengan alur yang telah dirancang.

Integrasi antar modul berjalan dengan baik tanpa ditemukan konflik data, serta histori perubahan stok dapat dicatat dengan lengkap.

Tabel 6.1. Hasil Evaluasi Pengujian

| Kriteria Pengujian | Hasil Pengujian |
|--|------------------------|
| Sistem dapat menampilkan antarmuka dengan baik | Terpenuhi |
| Validasi input pada form entri data | Terpenuhi |
| Penyimpanan data gudang | Terpenuhi |
| Penyimpanan data penyesuaian stok | Terpenuhi |
| Perhitungan stok dengan konversi satuan | Terpenuhi |
| Validasi stok sebelum pengurangan | Terpenuhi |
| Penyimpanan Purchase Request | Terpenuhi |
| Proses approval Purchase Request | Terpenuhi |
| Keterkaitan PR ke PO | Terpenuhi |
| Penerimaan barang dan update stok | Terpenuhi |

| | |
|---------------------------------------|-----------|
| Konsistensi data transaksi concurrent | Terpenuhi |
| Histori perubahan stok | Terpenuhi |

[Halaman ini sengaja dikosongkan]

BAB VII

KESIMPULAN DAN SARAN

7.1. Kesimpulan

Berdasarkan hasil analisis, perancangan, implementasi, serta pengujian sistem pada aplikasi Clusteria, diperoleh kesimpulan sebagai berikut:

- a. Perancangan sistem manajemen dan inventaris properti yang terstruktur dan terintegrasi dapat dilakukan dengan pendekatan modular, relasional, dan alur proses bisnis end-to-end pada satu platform aplikasi.
- b. Implementasi sistem yang mampu mengelola data properti dan inventaris secara efisien dan real-time dapat dicapai melalui integrasi transaksi antar modul, validasi operasional, serta sinkronisasi pembaruan data stok pada setiap proses bisnis.

7.2. Saran

Adapun beberapa saran untuk pengembangan sistem ke depannya adalah sebagai berikut:

- a. Melakukan optimasi performa query dan penambahan indeks pada kolom yang sering digunakan untuk pencarian, filter, dan join transaksi agar waktu respons tetap stabil saat volume data meningkat.
- b. Menambahkan mekanisme caching untuk data referensi yang sering diakses, sehingga beban akses ke basis data dapat dikurangi.
- c. Mengembangkan dashboard monitoring operasional yang menampilkan indikator utama, seperti pergerakan stok, status Purchase Request dan Purchase Order, dan progres pengadaan secara ringkas.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] Laravel, “Laravel Documentation,” [Daring]. Tersedia: <https://laravel.com/docs>. [Diakses: 13 Februari 2026].
- [2] Oracle, “MySQL Documentation,” [Daring]. Tersedia: <https://dev.mysql.com/doc/>. [Diakses: 19 Februari 2026].
- [3] Tailwind Labs, “Tailwind CSS Documentation,” [Daring]. Tersedia: <https://tailwindcss.com/docs/>. [Diakses: 23 Februari 2026].
- [4] Laravel, “Middleware Laravel Documentation,” [Daring]. Tersedia: <https://laravel.com/docs/13.x/middleware>. [Diakses: 1 Maret 2026].

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS I

Nama : Muhammad Rizqi Ar Rasyid
Tempat, Tanggal Lahir : Malang, 15 Juni 2005
Jenis Kelamin : Laki-laki
Telepon : +6285227765761
Email : rizqiarrasyid08@gmail.com

AKADEMIS

Kuliah : Departemen Teknik Informatika –
FTEIC , ITS
Angkatan : 2023
Semester : 6 (Enam)

BIODATA PENULIS II

Nama : Fadaukas Daffa Tajuddin
Tempat, Tanggal Lahir : Kediri, 27 September 2004
Jenis Kelamin : Laki-laki
Telepon : +6282249118184
Email : fadaukasdaffa04@gmail.com

AKADEMIS

Kuliah : Departemen Teknik Informatika –
FTEIC , ITS
Angkatan : 2023
Semester : 6 (Enam)