

**PENGEMBANGAN SISTEM INFORMASI *INVENTORY CONTROL* DAN
DATA ADMINISTRASI
(STUDI KASUS PADA BAGIAN LAYANAN KESEHATAN DPR-RI)
(Skripsi)**

Oleh

**FAIZ RIDHO UTOMO
NPM 2115061037**



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2025**

**PENGEMBANGAN SISTEM INFORMASI *INVENTORY CONTROL* DAN
DATA ADMINISTRASI (STUDI KASUS PADA BAGIAN LAYANAN
KESEHATAN DPR-RI)**

Oleh

FAIZ RIDHO UTOMO

Skripsi

**Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA TEKNIK**

Pada

**Program Studi Teknik Informatika
Jurusan Teknik Elektro
Fakultas Teknik Universitas Lampung**



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2025**

ABSTRAK

PENGEMBANGAN SISTEM INFORMASI *INVENTORY CONTROL* DAN DATA ADMINISTRASI STUDI KASUS PADA BAGIAN LAYANAN KESEHATAN DPR-RI

Oleh

FAIZ RIDHO UTOMO

Pengelolaan inventaris dan data administrasi secara manual berpotensi menyebabkan tidak tepatnya pencatatan serta keterlambatan dalam proses pelaporan. Penelitian ini bertujuan untuk merancang dan membangun sistem informasi terintegrasi guna mendukung pengelolaan barang dan data administrasi pada Bagian Layanan Kesehatan DPR-RI. Metode pengembangan yang digunakan adalah *Rapid Application Development* (RAD), yang mencakup tahapan perencanaan kebutuhan, perancangan antarmuka, pembangunan sistem, dan implementasi. Sistem dikembangkan menggunakan kerangka kerja Laravel dan basis data MySQL dengan *role admin* dan *employee* serta memiliki fitur utama diantaranya manajemen data barang, pengajuan cuti, pengarsipan surat, serta laporan. Evaluasi sistem dilakukan dengan metode *system usability scale* untuk menguji tahapan desain, *Black Box* untuk menguji fungsionalitas dan *User Acceptance Testing* (UAT) untuk menilai tingkat penerimaan pengguna. Hasil pengujian fungsional dengan dua iterasi, dengan iterasi pertama 94,37% iterasi kedua terdapat peningkatan, dengan tingkat keberhasilan 100% pada pengujian *Black Box* dan skor rata-rata UAT sebesar 86%. yang menunjukkan penerimaan pengguna yang sangat baik. Sistem mampu meningkatkan proses inventaris barang dan data administrasi.

Kata kunci: Sistem Informasi, Inventaris, Administrasi, Laravel, RAD, *Black Box*

ABSTRAK

PENGEMBANGAN SISTEM INFORMASI *INVENTORY CONTROL* DAN DATA ADMINISTRASI STUDI KASUS PADA BAGIAN LAYANAN KESEHATAN DPR-RI

Oleh

FAIZ RIDHO UTOMO

Manual management of inventory and administrative data has the potential to cause recording inaccuracies and delays in the reporting process. This research aims to design and develop an integrated information system to support goods management and administrative data at the Health Services Division of the Indonesian House of Representatives (DPR-RI). The development method used is Rapid Application Development (RAD), which includes the stages of requirements planning, interface design, system development, and implementation. The system was developed using the Laravel framework and MySQL database with admin and employee roles, featuring key functionalities including goods data management, leave application, document archiving, and reporting. System evaluation was conducted using the System Usability Scale method to test the design phase, Black Box testing for functionality testing, and User Acceptance Testing (UAT) to assess user acceptance levels. The functional testing results showed two iterations, with the first iteration achieving 94.37% and the second iteration showing improvement, reaching a 100% success rate in Black Box testing and an average UAT score of 86%, indicating very good user acceptance. The system successfully improved inventory goods and administrative data processes.

Keywords: Information System, Inventory, Administration, Laravel, RAD

Judul Skripsi

**: PENGEMBANGAN SISTEM INFORMASI
INVENTORY CONTROL DAN DA'
ADMINISTRASI STUDI KASUS PADA
BAGIAN LAYANAN KESEHATAN DPR-
RI**

Nama Mahasiswa

: Faiz Ridho Utomo

Nomor Pokok Mahasiswa

: 2115061037

Program Studi

: Teknik Informatika

Jurusan

: Teknik Elektro

Fakultas

: Teknik



Pembimbing Utama

Pembimbing Pendamping

Ir. Trisya Septiana, S.T., M.T., IPM.
NIP 199009212019032025

Resty Annisa, S.ST., M.Kom.
NIP 199008302019032019

2. Mengetahui

**Ketua Jurusan
Teknik Elektro**

**Ketua Program Studi
Teknik Informatika**

Herlinawati, S.T., M.T.
NIP 197103141999032001

Yessi Mulyani, S.T., M.T.
NIP 197312262000122001

MENGESAHKAN

1. Tim Penguji

Ketua : Ir. Trisya Septiana, S.T., M.T., IPM.

Sekretaris : Resty Annisa, S.ST., M.Kom.

Penguji : Mona Arif Muda, S.T., M.T.

2. Dekan Fakultas Teknik

Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc.

NIP 197509282001121002

Tanggal Lulus Ujian Skripsi: 4 Agustus 2025



SURAT PERNYATAAN

Dengan ini saya menyetujui bahwa skripsi berjudul "Pengembangan Sistem Informasi *Inventory Control* Dan Data Administrasi Studi Kasus Pada Bagian Layanan Kesehatan DPR-RI" sepenuhnya merupakan hasil karya saya sendiri. Apabila di kemudian hari terbukti pernyataan ini tidak benar, saya siap menerima sanksi sesuai dengan ketentuan hukum yang berlaku.

Bandar Lampung, 4 Agustus 2025

Penulis,



Faiz Ridho Utomo

NPM. 2115061037

RIWAYAT HIDUP



Penulis dilahirkan di Jakarta pada tanggal 02 Juni 2003 sebagai anak bungsu dari dua bersaudara, dari pasangan Bapak Kusman dan Ibu Paini. Penulis menyelesaikan jenjang pendidikan formal di MIN 18 Jakarta Timur pada tahun 2015, kemudian melanjutkan ke SMPN 184 Jakarta dan lulus pada tahun 2018, serta menamatkan pendidikan menengah atas di MAN 14 Jakarta pada tahun 2021. Pada tahun 2021, penulis diterima sebagai mahasiswa Program Studi Teknik Informatika Universitas Lampung melalui jalur seleksi SBMPTN. Selama masa perkuliahan, penulis aktif berpartisipasi dalam berbagai kegiatan, antara lain:

1. Mengikuti kegiatan Studi Independent Mitra Infinite Learning bagian Mobile *android* Kotlin developer batch 5 pada tahun 2023.
2. Penulis aktif sebagai anggota *English Society* Universitas Lampung dan telah terlibat dalam berbagai kegiatan, seperti penyelenggaraan *webinar* nasional, kepanitiaan.
3. Penulis aktif sebagai anggota Himpunan Mahasiswa Teknik Elektro Universitas Lampung kepengurusan tahun 2022-2023.

MOTTO

“Live as if you were to die tomorrow. Learn as if you were to live forever.”

(Mahatma Gandhi)

“The beginning is the most important part of the work.”

(Socrates)

“Don't end the book early because of a bad page”

(James)

“Semuanya tampak mustahil sampai hal itu diselesaikan”

(Penulis)

PERSEMBAHAN

Puji syukur penulis panjatkan ke hadirat Allah SWT yang telah melimpahkan rahmat, taufik, serta hidayah-Nya, saya sebagai penulis dapat menyelesaikan penelitian dan skripsi ini.

Kupersembahkan skripsi ini kepada:

Kedua Orang Tua

“Teruntuk kedua orang tua yang telah mendukung baik, dukungan moral maupun materil, yang telah memberikan doa sepanjang waktu kepada penulis”

SANWACANA

Puji syukur penulis panjatkan ke hadirat Allah SWT yang telah melimpahkan rahmat, taufik, serta hidayah-Nya, sehingga penulis dapat menyelesaikan penyusunan skripsi yang berjudul “Pengembangan Sistem Informasi *Inventory Control* Dan Data Administrasi Studi Kasus Pada Bagian Layanan Kesehatan DPR-RI”. Dalam proses penyusunan skripsi ini, penulis banyak menerima dukungan, bimbingan, serta bantuan baik secara moril maupun materil dari berbagai pihak. Oleh karena itu, dengan penuh rasa hormat dan terima kasih yang mendalam, penulis menyampaikan penghargaan sebesar-besarnya kepada:

1. Pintu surgaku ibunda tercinta yaitu ibu Paini yang telah melahirkan membesarkan dan memberikan kasih sayang dan cintanya kepada penulis, serta menjadi tempat berpulang ternyaman bagi penulis. Terima kasih atas doa yang beliau panjatkan tiada henti, sehingga dapat sampai titik ini mendapatkan gelar sarjana;
2. Teruntuk Ayah tercinta panutanku yaitu Bapak Kusman, terima kasih sudah berjuang untuk kehidupan penulis, sebagai panutan yang menjadi contoh terbaik sebagai laki – laki;
3. Saudari kandung perempuan penulis Alfianita Fadila yang telah memberikan semangat serta memberikan arti perjuangan menyelesaikan studi untuk mendapatkah gelar sarjana;
4. Bapak Dr. Eng. Helmy Fitriawan, S.T., M.Sc., selaku Dekan Fakultas Teknik Universitas Lampung;
5. Ibu Herlinawati, S.T., M.T., selaku Ketua Jurusan Teknik Elektro Universitas Lampung;
6. Ibu Yessi Mulyani, S.T., M.T. selaku Ketua Program Studi Teknik Informatika Universitas Lampung yang telah membantu proses kelancaran pengerjaan penelitian;

7. Ibu Ir. Trisya Septiana, S.T., M.T., IPM. selaku Pembimbing Utama yang telah membimbing dan memberikan saran kepada penulis selama mengerjakan penelitian dan penulisan skripsi;
8. Ibu Resty Annisa, S.ST., M.Kom. selaku Pembimbing Pendamping yang telah membimbing dan memberikan saran bagi penulis dalam menyelesaikan penelitian dan penulisan skripsi;
9. Bapak Mona Arif Muda, S.T., M.T. selaku Penguji yang telah memberikan berbagai saran dan masukan yang membangun terhadap penelitian ini;
10. dr Dian Mardiyani serta jajaran pegawai Yankes DPR-RI yang telah memberikan kesempatan untuk melakukan penelitian dan bersedia meluangkan waktu untuk menjadi penguji sistem;
11. Kepada teman - teman penulis di bangku SMA yaitu, Yayan, Aleandra, Hani, Kinanti, Almeyra, Alfina. Terima kasih atas dukungan dan tempat saling bercerita ketika kembali ke lingkungan rumah;
12. Kepada seseorang yang tidak kalah penting kehadirannya, Erika Dwi Utami. Terima kasih telah menjadi bagian dalam proses perjalanan penulis menyusun skripsi, menemani, mendukung, serta mendengarkan keluh kesan dan meyakinkan penulis untuk pantang menyerah;
13. *Last but not least*, terima kasih untuk diri sendiri, karena telah mampu berjuang sejauh ini untuk mengejar gelar sarjana, yang telah mampu mengendalikan diri dari berbagai tekanan dan tak pernah memutuskan untuk menyerah sesulit apapun prosesnya.

Penulis berharap agar laporan ini dapat menjadi referensi bagi pengembangan keilmuan di bidang teknik informatika. Oleh karena itu, semoga penelitian ini bermanfaat bagi yang membacanya.

Bandar Lampung, 4 Agustus 2025

Penulis,



Faiz Ridho Utomo

DAFTAR ISI

| | Halaman |
|--|---------|
| ABSTRAK | i |
| PERSEMBAHAN | viii |
| SANWACANA | ix |
| DAFTAR GAMBAR | xiv |
| DAFTAR TABEL | xviii |
| I. PENDAHULUAN | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Rumusan Masalah | 2 |
| 1.3 Tujuan Penelitian | 3 |
| 1.4 Manfaat Penelitian | 3 |
| 1.5 Batasan Masalah | 3 |
| 1.6 Sistematika Penulisan | 4 |
| II. TINJAUAN PUSTAKA | 5 |
| 2.1 Dasar Teori | 5 |
| 2.1.1 Sistem Informasi | 5 |
| 2.1.2 Inventaris | 5 |
| 2.1.3 Data administrasi | 5 |
| 2.1.4 <i>Rapid Application Development</i> | 6 |
| 2.1.5 <i>Unified Modeling Language (UML)</i> | 7 |
| 2.1.6 <i>Black box testing</i> | 8 |
| 2.1.7 <i>User Acceptance Testing (UAT)</i> | 8 |
| 2.1.8 <i>Visual Studio Code</i> | 8 |
| 2.1.9 Git | 9 |
| 2.1.10 <i>Github</i> | 9 |
| 2.1.11 XAMPP | 9 |
| 2.1.12 <i>Database MySQL</i> | 10 |
| 2.1.13 <i>Laravel</i> | 10 |
| 2.1.14 <i>PHP(Hypertext Preprocessor)</i> | 11 |

| | | |
|--------|---|----|
| 2.1.15 | <i>System Usability Scale</i> | 11 |
| 2.2 | Penelitian Terkait..... | 12 |
| 2.2.1 | <i>Evolving Trends in Open-Source RDBMS: Performance, Scalability, and Security Insights</i> | 12 |
| 2.2.2 | <i>Design and Implementation of Inventory Management System for University</i> | 12 |
| 2.2.3 | <i>Black box testing with Equivalence Partitions Techniques in Transcrop Applications</i> | 12 |
| 2.2.4 | <i>Applying agile practices to avoid chaos in User Acceptance Testing: A case study</i> | 13 |
| 2.2.5 | <i>Implementation of Rapid Application Development Method in the Development of Geographic Information Systems of Industrial Centers</i> | 13 |
| 2.2.6 | Pengujian Fungsionalitas Sistem Inventaris Barang Pada Sekolah Menengah Kejuruan Citra Borneo Menggunakan <i>Black box testing</i> | 14 |
| 2.2.7 | Analisis Perbandingan Unjuk Kerja <i>Database SQL</i> dan <i>Database NoSQL</i> Untuk Mendukung <i>Era Big Data</i> | 14 |
| 2.2.8 | Sistem Manajemen Pelayanan Pelanggan Menggunakan <i>PHP</i> Dan <i>MySQL</i> (Studi Kasus pada Toko Surya) | 15 |
| 2.2.9 | Rancang Bangun Aplikasi Web Sekolah Menggunakan <i>Uml (Unified Modeling Language)</i> Dan Bahasa Pemrograman <i>PHP (Php Hypertext Preprocessor)</i> | 15 |
| 2.2.10 | Evaluasi Kegunaan Aplikasi Akhlaqu Menggunakan Metode <i>System Usability Scale (SUS)</i> dan Implementasi pada <i>MongoDB</i> | 16 |
| III. | METODOLOGI PENELITIAN..... | 17 |
| 3.1 | Waktu dan Tempat Penelitian..... | 17 |
| 3.2 | Alat Penelitian | 18 |
| 3.3 | Tahapan Penelitian..... | 19 |
| 3.3.1 | Analisis Kebutuhan..... | 20 |
| 3.4.1 | <i>Construction(implementasi)</i> | 57 |
| 3.4.2 | <i>Cutover(pengujian)</i> | 57 |
| IV. | HASIL DAN PEMBAHASAN..... | 59 |

| | |
|---|-----|
| 4.1. <i>Construction</i> (implementasi) Portal Registrasi Inventaris dan Manajemen administrasi | 59 |
| 4.2. Hasil implementasi PRIMA (portal registrasi inventaris dan administrasi) | 65 |
| 4.3. <i>Cutover</i> (pengujian) | 106 |
| 4.3.1. <i>User Acceptence testing</i> | 106 |
| 4.3.2. <i>Black box testing</i> | 111 |
| 4.4. <i>Deployment</i> | 137 |
| 4.4.1. <i>Arsitektur Deployment</i> | 137 |
| 4.4.2. <i>Spesifikasi server shared hosting Niagahoster</i> | 138 |
| 4.4.3. <i>Estimasi penyimpanan</i> | 138 |
| V. KESIMPULAN DAN SARAN | 143 |
| 5.1. Kesimpulan | 143 |
| 5.2. Saran..... | 144 |
| DAFTAR PUSTAKA | 145 |

DAFTAR GAMBAR

| | Halaman |
|---|---------|
| Gambar 2. 1 <i>Metode Rapid Application Development</i> [8] | 6 |
| Gambar 2. 2 <i>Arsitektur Kerangka Kerja Laravel</i> [17]..... | 11 |
| Gambar 3. 1 Tahapan Penelitian | 19 |
| Gambar 3. 2 <i>Business process</i> saat ini | 24 |
| Gambar 3. 3 <i>Business process Recommendation</i> (digitalisasi) | 25 |
| Gambar 3. 4 Alur Sistem..... | 26 |
| Gambar 3. 5 <i>Arsitektur Sistem</i> | 27 |
| Gambar 3.6 <i>Use Case</i> | 28 |
| Gambar 3. 7 <i>Activity diagram login</i> | 29 |
| Gambar 3. 8 <i>activity diagram</i> barang masuk | 30 |
| Gambar 3. 9 <i>Activity diagram</i> barang keluar | 31 |
| Gambar 3. 10 <i>Activity diagram</i> pengajuan cuti..... | 32 |
| Gambar 3. 11 <i>Activity diagram</i> validasi cuti..... | 33 |
| Gambar 3. 12 <i>Activity diagram</i> arsip surat..... | 34 |
| Gambar 3. 13 <i>Activity diagram</i> pengembalian..... | 35 |
| Gambar 3. 14 <i>Activity diagram</i> laporan barang masuk..... | 36 |
| Gambar 3. 15 <i>Activity diagram</i> laporan barang masuk..... | 37 |
| Gambar 3. 16 <i>Activity diagram</i> laporan stok barang..... | 38 |
| gambar 3. 17 <i>Entity Relational Diagram</i> Dashboard..... | 39 |
| Gambar 3. 18 <i>Desain mockup</i> halaman landing page | 41 |
| Gambar 3. 19 <i>Desain mockup</i> halaman login..... | 42 |
| Gambar 3. 20 Halaman utama <i>dashboard</i> | 42 |
| Gambar 3. 21 Jenis barang | 43 |
| Gambar 3. 22 Satuan barang | 44 |
| Gambar 3. 23 Merk barang | 44 |
| Gambar 3. 24 Barang | 45 |

| | |
|--|----|
| Gambar 3. 25 Desain mockup halaman notifikasi email | 46 |
| Gambar 3. 26 Halaman peminjam | 46 |
| Gambar 3. 27 Desain <i>mockup</i> halaman notifikasi..... | 47 |
| Gambar 3. 28 Halaman pengajuan cuti | 48 |
| Gambar 3. 29 Halaman validasi cuti | 48 |
| Gambar 3. 30 Halaman pemasok | 49 |
| Gambar 3. 31 Halaman transaksi masuk..... | 50 |
| Gambar 3. 32 Halaman transaksi masuk..... | 50 |
| Gambar 3. 33 Halaman barang masuk | 51 |
| Gambar 3. 34 Halaman laporan barang keluar..... | 51 |
| Gambar 3. 35 Halaman laporan stok barang | 52 |
| Gambar 3. 36 Halaman pengaturan pengguna | 53 |
| Gambar 3. 37 Halaman Profile | 53 |
| gambar 3. 38 Desain mockup halaman pengembalian barang | 56 |
| Gambar 4. 1 Struktur Folder | 59 |
| Gambar 4. 2 Struktur file manajemen inventaris | 60 |
| Gambar 4. 3 Struktur folder <i>Http/controller</i> | 62 |
| Gambar 4. 4 Struktur Folder Models | 63 |
| Gambar 4. 5 Struktur <i>file resource/view</i> | 64 |
| Gambar 4.6 Hasil halaman landing page | 65 |
| Gambar 4.7 Hasil implementasi <i>code landing page</i> | 66 |
| Gambar 4.8 Hasil implementasi halaman <i>login</i> | 67 |
| Gambar 4. 9 Hasil implementasi <i>code login</i> | 67 |
| Gambar 4. 10 Hasil implementasi halaman utama <i>role employee dashboard</i> | 68 |
| Gambar 4. 11Hasil implementasi <i>code</i> halaman utama <i>dashboard</i> | 69 |
| Gambar 4. 12 Hasil implementasi halaman jenis barang <i>role employee</i> | 69 |
| Gambar 4. 13 Hasil implementasi <i>code</i> jenis barang | 70 |
| Gambar 4. 14 Hasil implementasi halaman satuan barang <i>role employee</i> | 71 |
| Gambar 4. 15 Hasil implementasi <i>code</i> satuan barang | 71 |
| Gambar 4. 16 Hasil implementasi halaman merek barang <i>role employee</i> | 72 |
| Gambar 4. 17 Hasil implementasi halaman barang <i>role employee</i> | 72 |
| Gambar 4. 18 Hasil implementasi <i>code</i> barang | 73 |

| | |
|---|----|
| Gambar 4. 19 Hasil implementasi halaman jenis surat <i>role employee</i> | 73 |
| Gambar 4. 20 Hasil implementasi halaman satuan surat <i>role employee</i> | 74 |
| Gambar 4. 21 Hasil implementasi halaman peminjam <i>role employee</i> | 74 |
| Gambar 4. 22 Hasil implementasi halaman pengirim surat <i>role employee</i> | 75 |
| Gambar 4. 23 Hasil implementasi halaman distributor <i>role employee</i> | 75 |
| Gambar 4. 24 Hasil transaksi barang masuk <i>role employee</i> | 76 |
| Gambar 4. 25 Hasil implementasi <i>code</i> transaksi masuk..... | 76 |
| Gambar 4. 26 Hasil implementasi transaksi barang keluar..... | 77 |
| Gambar 4. 27 Hasil Implementasi <i>code</i> transaksi barang keluar | 78 |
| Gambar 4. 28 Hasil implementasi surat masuk..... | 79 |
| Gambar 4. 29 Hasil implementasi halaman surat keluar <i>role employee</i> | 80 |
| Gambar 4. 30 Hasil implementasi halaman pengajuan cuti <i>role employee</i> | 80 |
| Gambar 4. 31 Hasil implementasi status pengajuan cuti disetujui..... | 81 |
| Gambar 4. 32 Hasil implementasi <i>code</i> pengajuan cuti | 81 |
| Gambar 4. 33 Hasil implementasi halaman pengembalian <i>role employee</i> | 82 |
| Gambar 4. 34 Hasil implementasi laporan barang masuk..... | 82 |
| Gambar 4. 35 Hasil implementasi laporan barang keluar <i>role employee</i> | 83 |
| Gambar 4. 36 Hasil implementasi laporan stok barang <i>role employee</i> | 83 |
| Gambar 4. 37 Hasil implementasi <i>code</i> laporan stok barang..... | 84 |
| Gambar 4. 38 Hasil implementasi laporan surat masuk <i>role employee</i> | 84 |
| Gambar 4. 39 Hasil implementasi <i>code</i> laporan surat masuk | 85 |
| Gambar 4. 40 Hasil implementasi laporan surat keluar <i>role employee</i> | 86 |
| Gambar 4. 41 Hasil implementasi pengaturan <i>profile</i> | 86 |
| Gambar 4. 42 Hasil implementasi <i>login</i> | 87 |
| Gambar 4. 43 Hasil implementasi <i>code</i> login | 88 |
| Gambar 4. 44 Hasil implementasi dasbor <i>role admin</i> | 88 |
| Gambar 4. 45 Hasil implementasi jenis barang <i>role admin</i> | 89 |
| Gambar 4. 46 Hasil implementasi satuan barang <i>role admin</i> | 89 |
| Gambar 4. 47 Hasil implementasi <i>code</i> satuan barang | 90 |
| Gambar 4. 48 Hasil implementasi merek barang <i>role admin</i> | 90 |
| Gambar 4. 49 Hasil implementasi barang <i>role admin</i> | 91 |
| Gambar 4. 50 Hasil implementasi jenis surat roler admin | 91 |

| | |
|---|-----|
| Gambar 4. 51 Hasil implementasi satuan surat <i>role admin</i> | 92 |
| Gambar 4. 52 Hasil implementasi peminjam <i>role admin</i> | 93 |
| Gambar 4. 53 Hasil implementasi halaman pengirim surat <i>role admin</i> | 93 |
| Gambar 4. 54 Hasil implementasi halaman distributor <i>role admin</i> | 94 |
| Gambar 4. 55 Hasil implementasi halaman barang masuk <i>role admin</i> | 94 |
| Gambar 4. 56 Hasil implementasi halaman transaksi keluar <i>role admin</i> | 95 |
| Gambar 4. 57 Hasil implementasi <i>code</i> transaksi barang keluar | 96 |
| Gambar 4. 58 Hasil implementasi surat masuk <i>role admin</i> | 96 |
| Gambar 4. 59 Hasil implementasi halaman pengajuan cuti <i>role admin</i> | 97 |
| Gambar 4. 60 Hasil implementasi halaman <i>leave validation</i> <i>role admin</i> | 98 |
| Gambar 4. 61 Hasil implementasi halaman notifikasi email <i>role admin</i> | 98 |
| Gambar 4. 62 Hasil implementasi kode validasi cuti..... | 99 |
| Gambar 4. 63 Hasil implementasi kode template email validasi cuti..... | 100 |
| Gambar 4. 64 Hasil implementasi halaman pengembalian <i>role admin</i> | 100 |
| Gambar 4. 65 Hasil implementasi halaman laporan barang masuk <i>role admin</i> .. | 101 |
| Gambar 4. 66 Hasil implementasi kode laporan barang masuk..... | 101 |
| Gambar 4. 67 Hasil implementasi laporan barang keluar <i>role admin</i> | 102 |
| Gambar 4. 68 Hasil implementasi kode laporan barang keluar | 102 |
| Gambar 4. 69 Hasil implementasi halaman laporan barang keluar <i>role admin</i> .. | 103 |
| Gambar 4. 70 Hasil implementasi kode laporan surat masuk..... | 104 |
| Gambar 4. 71 Hasil implementasi halaman laporan surat keluar <i>role admin</i> | 104 |
| Gambar 4. 72 Hasil implementasi halaman pengaturan pengguna <i>role admin</i> ... | 105 |
| Gambar 4. 73 Hasil implementasi kode pengaturan pengguna..... | 105 |
| Gambar 4. 74 Arsitektur <i>Deployment</i> | 137 |
| Gambar 4. 75 Mengisi informasi kepemilikan hosting..... | 141 |
| Gambar 4. 76 Mengisi informasi database niaga hoster hosting baru | 141 |
| Gambar 4. 77 <i>Upload</i> file ke dalam hpanel niaga hoster | 142 |
| Gambar 4. 78 <i>Website</i> primasiyankes.id | 142 |

DAFTAR TABEL

Halaman

| | |
|--|-----|
| Tabel 3. 1 Waktu Penelitian..... | 17 |
| Tabel 3. 2 Alat dan bahan..... | 18 |
| Tabel 3. 3 Borang Pertanyaan fungsionalitas dan non fungsional sistem | 20 |
| Tabel 3. 4 Kebutuhan fungsional | 22 |
| Tabel 3. 5 Kebutuhan non fungsional | 23 |
| Tabel 3. 6 Definisi <i>use case</i> | 28 |
| Tabel 3. 7 Hasil Perhitungan <i>system usability scale</i> | 54 |
| Tabel 3. 8 Skala Interpretasi Hasil Skor SUS [29]..... | 55 |
| tabel 4. 1 pertanyaan <i>user acceptance testing</i> | 106 |
| tabel 4. 2 Hasil pengujian UAT <i>usability</i> | 108 |
| tabel 4. 3 Hasil pengujian UAT <i>functionality</i> | 109 |
| tabel 4. 4 Hasil pengujian uat <i>efficiency</i> | 110 |
| tabel 4. 5 kategori penilaian skor [30], [31]..... | 111 |
| tabel 4. 6 Ringkasan Pengujian login dan logout halaman <i>dashboard</i> | 112 |
| tabel 4. 7 Ringkasan Pengujian manajemen pengguna iterasi ke-1 | 113 |
| tabel 4. 8 Ringkasan Pengujian manajemen pengguna iterasi ke-2 | 113 |
| tabel 4. 9 Ringkasan Pengujian manajemen inventaris halaman <i>dashboard</i> | 114 |
| tabel 4. 10 Hasil <i>black box testing</i> Ringkasan Pengujian manajemen surat | 116 |
| tabel 4. 11 Hasil <i>black box testing</i> manajemen cuti iterasi ke-1 | 117 |
| tabel 4. 12 Hasil <i>black box testing</i> manajemen cuti iterasi ke-2 | 118 |
| tabel 4. 13 Ringkasan Pengujian Laporan barang..... | 119 |
| tabel 4. 14 Ringkasan cetak laporan surat..... | 120 |
| tabel 4. 15 Ringkasan cetak laporan surat..... | 121 |
| tabel 4. 16 Ringkasan cetak <i>edit profile</i> | 122 |
| tabel 4. 17 Total iterasi 1 <i>black box testing</i> | 122 |
| tabel 4. 18 Total iterasi ke-2 <i>black box testing</i> | 125 |

| | |
|--|-----|
| tabel 4. 19 Tabel spesifikasi <i>deployment</i> | 138 |
| tabel 4. 20 Estimasi penyimpanan satuan surat..... | 138 |
| tabel 4. 21 Estimasi penyimpanan surat masuk | 139 |
| tabel 4. 22 Estimasi penyimpanan surat keluar..... | 139 |
| tabel 4. 23 Estimasi total penyimpanan | 139 |
| tabel 4. 24 <i>Breakdown</i> Estimasi penyimpanan..... | 140 |

I. PENDAHULUAN

1.1 Latar Belakang

Pengelolaan inventaris dan data administrasi kantor yang baik dan tercatat secara rapi merupakan hal yang sangat penting bagi kelancaran operasional organisasi. Masih banyak lembaga yang menghadapi tantangan besar dalam hal ini [1]. Menurut Hasil laporan audit BPK pada tahun 2022 ditemukan 60-75% dari instansi yang diaudit menunjukkan kelemahan dalam pencatatan inventaris dan data administrasi berdasarkan temuan yang menunjukkan ketidakpatuhan terhadap ketentuan dan pengendalian yang tidak memadai data diambil berdasarkan (penyusunan laporan keuangan, pengelolaan pendapatan, belanja dan aset, dan piutang aset). Akibatnya pengelolaan barang yang tidak baik sering kali mengarah pada kelalaian administratif, yang memperburuk kualitas layanan publik dan mengurangi produktivitas. Penelitian mengenai pelayanan publik di Kota Pekanbaru, ditemukan bahwa sistem administrasi yang buruk dan tidak transparan, serta kelalaian dalam pengelolaan data administrasi, menyebabkan ketidakpuasan masyarakat dan meningkatkan potensi mal administrasi [2].

Hal tersebut juga dialami dalam bagian Layanan Kesehatan DPR-RI hingga saat ini masih mengandalkan proses manual dalam pencatatan inventaris dan data administrasi, yang memicu kesalahan operasional. Setiap permintaan barang maupun perlengkapan kantor dicatat terlebih dahulu pada formulir kertas yang kemudian input ulang ke dalam *Microsoft Excel* terpisah. Begitu pula saat penerimaan barang data nama, jumlah, dan tanggal terima dicatat secara manual, alur serupa terjadi untuk pencatatan barang keluar: formulir kertas pengajuan dibuat ulang di *Excel*. Dari sisi administrasi pegawai, masalah serupa terlihat pada proses pengajuan cuti, nota dinas, surat keluar, surat masuk. Formulir pengajuan cuti dicetak dan ditandatangani oleh kepala bagian, ketergantungan pada dokumen terpisah ini memaksa staf melakukan pengecekan silang ke beberapa berkas saat membutuhkan informasi, sehingga laporan bulanan tentang stok dan administrasi cuti memerlukan proses pelaporan yang lama. Untuk mengatasi permasalahan tersebut, diperlukan implementasi digitalisasi pencatatan barang dan administrasi

yang terintegrasi. Sistem digitalisasi data inventaris dan administrasi ini diharapkan dapat mengurangi kesalahan proses, meminimalkan kesalahan operasional, dan meningkatkan efisiensi kerja. Proses digitalisasi ini, meliputi pencatatan inventaris dan administrasi dapat dilakukan dalam satu sistem *website*, sehingga memudahkan akses informasi dan mempercepat proses pelaporan. [3]. Salah satu metode yang dapat diterapkan dalam pengembangannya adalah penerapan metode *Rapid Application Development* (RAD) dalam pengembangan sistem informasi inventaris barang dan data administrasi dapat menjadi solusi yang efektif. Metode RAD memungkinkan pengembangan sistem yang lebih cepat dan adaptif terhadap kebutuhan pengguna, sehingga proses perencanaan, perancangan, dan implementasi dapat diselesaikan dalam waktu yang relatif singkat [4]. Pengembangan sistem informasi inventaris dan data administrasi menggunakan metode RAD tidak hanya memperbaiki pengelolaan aset dan administrasi, tetapi juga secara signifikan meningkatkan produktivitas, akurasi data, dan kualitas layanan organisasi secara keseluruhan.

1.2 Rumusan Masalah

Adapun rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Apa saja fitur dan fungsionalitas yang dibutuhkan pengguna untuk memudahkan pengelolaan inventaris barang dan data administrasi di bagian Layanan Kesehatan DPR RI?
2. Bagaimana pengujian *black box testing* dan *User Acceptance Testing* dapat membantu untuk pengembangan sistem informasi *inventory control* dan data administrasi di layanan bagian Kesehatan DPR-RI ?
3. Bagaimana penerapan tahapan metode RAD (*Requirement Planning, User Design, Construction, Cutover*) dalam pengembangan sistem informasi inventaris dan administrasi ini?

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Pengembangan sistem informasi *inventory control* dan administrasi data untuk membantu proses administrasi.
2. Menguji fungsionalitas dan tahapan desain pada sistem informasi *inventory control* dan administrasi data supaya sesuai dengan kebutuhan pengguna.

1.4 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah sebagai berikut:

1. Membantu meningkatkan pencatatan data administrasi di bagian Layanan Kesehatan DPR-RI.
2. Membantu dalam transparansi dan mengurangi pemborosan, serta akurasi dalam pengelolaan barang, sehingga kualitas pelayanan kesehatan dapat terjaga dengan lebih baik.
3. Berkontribusi dalam digitalisasi pelaporan inventaris barang di bagian layanan kesehatan DPR-RI

1.5 Batasan Masalah

Adapun batasan masalah dari penelitian ini adalah sebagai berikut:

1. Sistem ini menggunakan *framework Laravel* dan *database MySQL*. Proses tahap *deployment* diimplementasikan pada *server shared hosting*.
2. Sistem ini berfokus pada implementasi fitur-fitur untuk pengguna dengan peran (*role-based access control*) yaitu *role admin*, dan *employee* untuk memastikan menambahkan data barang serta data administrasi perkantoran.
3. Pengujian sistem dibatasi pada pengujian fungsional menggunakan *black box testing* dan menggunakan keamanan bawaan *framework Laravel*

1.6 Sistematika Penulisan

BAB I: PENDAHULUAN

Bab ini membahas latar belakang kebutuhan digitalisasi di Bagian Layanan Kesehatan DPR-RI, terutama dalam hal pengelolaan inventaris barang dan administrasi seperti pengajuan cuti serta arsip surat. Bab ini juga memuat rumusan masalah yang mencakup kebutuhan fitur dan bagaimana sistem diuji, serta batasan penelitian (hanya meliputi admin dan pegawai di unit terkait). Disusun pula sistematika penulisan skripsi yang membagi isi menjadi lima bab.

BAB II: TINJAUAN PUSTAKA

Berisi kajian teori yang menjadi dasar pengembangan sistem, mencakup konsep *inventory control*, sistem informasi administrasi, dan teknologi web (Laravel, MySQL). Di jelaskan pula prinsip dan tahapan dari *Rapid Application Development* (RAD), yang meliputi: *planning, design, construction, dan cutover*.

BAB III: METODE PENELITIAN

Bab ini menjelaskan metode penelitian yang digunakan yaitu metode *Rapid Application Development*. Bab ini berisi mengenai waktu penelitian, tempat penelitian, alat penelitian serta metode dan tahap yang dilakukan dalam penelitian ini.

BAB IV: HASIL DAN PEMBAHASAN

Bab ini berisi hasil rancangan dari sistem dengan tahapan metode *Rapid Application development* yaitu *construction* dan *cutover* serta, penjelasan hasil pengujian aplikasi yang telah dirancang.

BAB V: KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan yang diperoleh dari hasil penelitian yang telah dilakukan serta berisi saran yang dapat digunakan sebagai acuan untuk penelitian selanjutnya di masa yang datang.

II. TINJAUAN PUSTAKA

2.1 Dasar Teori

2.1.1 Sistem Informasi

Sistem informasi adalah suatu sistem yang dirancang untuk mengumpulkan, memproses, menyimpan, dan mendistribusikan informasi guna mendukung pengambilan keputusan dan operasional dalam suatu organisasi. Sistem ini terdiri dari berbagai komponen, termasuk perangkat keras, perangkat lunak, data, jaringan komunikasi, serta prosedur yang digunakan untuk mengelola informasi memiliki fungsi khusus dalam mendukung aktivitas organisasi [5].

2.1.2 Inventaris

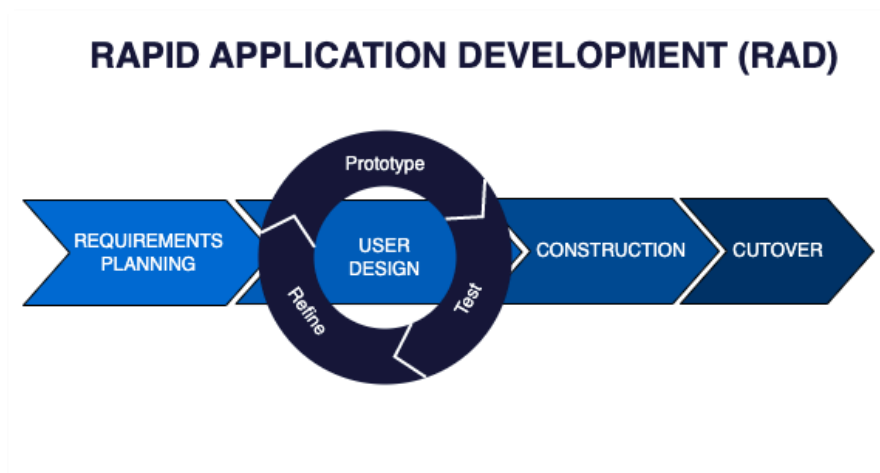
Inventaris adalah sistem pencatatan dan pengelolaan aset atau barang yang dimiliki oleh suatu organisasi atau perusahaan untuk memastikan penggunaan yang efisien dan pemeliharaan yang tepat. Proses ini mencakup identifikasi, pencatatan, pengawasan, serta pemantauan stok barang agar dapat digunakan secara optimal sesuai kebutuhan operasional. Inventaris yang baik memungkinkan perusahaan mengurangi pemborosan, mengoptimalkan pengadaan barang, serta memastikan ketersediaan sumber daya yang diperlukan [6].

2.1.3 Data administrasi

Administrasi perkantoran merupakan sistem pengelolaan informasi dan sumber daya dalam suatu organisasi untuk memastikan kelancaran operasional dan pencapaian tujuan bisnis. Ini mencakup perencanaan, pengorganisasian, pengarahan, dan pengendalian berbagai aktivitas seperti manajemen dokumen, komunikasi internal dan eksternal, serta pengelolaan data dan teknologi informasi. Efektivitas administrasi perkantoran sangat bergantung pada sistem informasi yang digunakan untuk meningkatkan efisiensi dalam pengambilan keputusan dan koordinasi antar departemen [6].

2.1.4 *Rapid Application Development*

Metode *Rapid Application Development* (RAD) adalah suatu pendekatan dalam pengembangan perangkat lunak yang menekankan pada kecepatan dan efisiensi. Pendekatan ini berusaha untuk mempersingkat waktu siklus pengembangan dengan memanfaatkan prototipe dan iterasi yang cepat. RAD mengedepankan integrasi umpan balik dari pengguna secara langsung dalam proses pengembangan, sehingga pengembang dapat dengan cepat menanggapi perubahan kebutuhan yang muncul. [7].



Gambar 2. 1 Metode *Rapid Application Development* [8]

Adapun tahapan pengembangan *Rapid Application Development* sebagai berikut:

1. Rencana Kebutuhan (*Requirements Planning*)

Tahap yang pertama ini pengguna (*user*) dan *developer* saling bertemu untuk memecahkan permasalahan yang sedang terjadi. Dalam tahapan ini antara pengguna dan juga *developer* menentukan apa saja yang dibutuhkan untuk membuat sistem informasi, karena tahap ini merupakan keberhasilan sistem serta dapat menghindari kesalahan komunikasi antara pengguna dan *developer*.

2. Desain pengguna (*user design*)

Tahap ini merupakan tahap rancangan yang diusulkan sesuai dengan kebutuhan, berjalan sesuai dengan kebutuhan berjalan sesuai yang direncan dan diharapkan dapat menyelesaikan permasalahan yang sedang terjadi. Pada penelitian ini, desain sistem digambarkan menggunakan UML (*Unified Modeling Language*).

3. Pengembangan (*construction*)

Tahapan ini merupakan sebuah tahapan yang di mana sebagai developer memulai membuat sistem yang sudah direncan sebelumnya. Memulai menyusun suatu baris kode program (*coding*) untuk mengubah sistem desain menjadi sebuah sistem informasi *website* yang telah direncan agar dapat digunakan.

4. Implementasi (*Cutover*)

Tahapan ini merupakan sebuah tahapan pengujian dari kelancaran keseluruhan sistem yang dibangun, semua komponen perlu diuji secara menyeluruh menggunakan metode *Black box testing*, dalam tahapan pengujian ini bertujuan untuk dapat mengurangi risiko dari cacat sistem. *Black box testing* merupakan pengujian yang berfokus pada pengujian fungsional dari perangkat lunak yang dibangun [8].

2.1.5 *Unified Modeling Language* (UML)

Unified Modeling Language (UML) adalah sebuah bahasa yang digunakan untuk memvisualisasikan, menggambarkan, dan mendokumentasikan sistem berbasis objek. Penelitian ini, dua jenis diagram UML yang digunakan adalah *Use Case Diagram* dan *Activity diagram*. *Use Case Diagram* menggambarkan interaksi antara aktor dan sistem. Penggunaan *Use Case Diagram* memberikan manfaat dalam memahami alur interaksi tersebut dengan lebih jelas [9]. Ada beberapa diagram UML yang umum digunakan dalam pengembangan sistem, di antaranya:

- *Use Case Diagram*: Menggambarkan fungsionalitas yang diharapkan dari sistem serta interaksi antara aktor (manusia atau sistem lain) dan sistem itu sendiri.

- *Activity diagram*: Menunjukkan aliran aktivitas dalam sistem yang sedang berjalan.

2.1.6 *Black box testing*

Black box testing adalah metode pengujian perangkat lunak yang berfokus pada analisis input dan *output* tanpa melihat struktur internal atau kode programnya. Pengujian ini sering disebut sebagai pengujian berbasis spesifikasi karena hanya mengacu pada spesifikasi perangkat lunak dalam menentukan kasus uji. Keunggulan utama *black box testing* metode pengujian ini dapat dilakukan secara independen dari implementasi kode, memungkinkan pengujian dilakukan secara paralel dengan pengembangan sistem, serta tetap berguna meskipun terjadi perubahan implementasi. metode ini juga memiliki kekurangan, seperti kemungkinan adanya ambiguitas dalam pengujian serta celah yang tidak terdeteksi akibat keterbatasan dalam menguji skenario yang tidak tercantum dalam spesifikasi [10].

2.1.7 *User Acceptance Testing (UAT)*

User Acceptance Testing (UAT) adalah proses validasi untuk memastikan bahwa solusi yang dikembangkan dalam sistem perangkat lunak yang dirancang sudah sesuai dengan kebutuhan dan harapan dari pengguna akhir (*end user*) yang di mana *end user* tersebut adalah staf atau karyawan perusahaan yang memakai sistem kita buat, yang di mana UAT ini bertujuan untuk memastikan bahwa sistem yang dirancang sudah sesuai semua fungsi dari yang diharapkan dapat berjalan dengan baik, untuk memenuhi kriteria serta kebutuhan dari pengguna akhir. UAT juga memiliki peran untuk memastikan bahwa sistem yang dibuat tidak hanya berfungsi secara teknis, tetapi juga dapat diterima serta digunakan dengan baik oleh pengguna [11].

2.1.8 *Visual Studio Code*

Visual Studio Code (VSCode) adalah editor kode sumber terbuka yang dikembangkan oleh Microsoft dan telah menjadi salah satu pilihan utama bagi pengembang perangkat lunak. Keunggulan utama *VSCode* terletak pada

fleksibilitasnya yang tinggi, didukung oleh berbagai ekstensi yang memperkaya fungsionalitasnya, seperti dukungan untuk berbagai bahasa pemrograman, *debugging*, dan integrasi dengan sistem kontrol versi seperti Git [12].

2.1.9 Git

Git adalah sistem kontrol versi yang kuat dan fleksibel, dirancang untuk mendukung pengelolaan kode secara terdistribusi. Dengan Git, pengembang dapat dengan mudah membuat, mengelola, dan berbagi perubahan dalam kode sumber secara efisien. Salah satu keunggulan utama Git adalah sistem *branch* dan *merging* yang memungkinkan pengembangan paralel tanpa mengganggu versi utama proyek. Git memiliki fitur *commit tracking*, *revert* perubahan, dan integrasi dengan layanan seperti GitHub dan GitLab menjadikan Git sebagai alat yang sangat berguna dalam kolaborasi tim dan pengelolaan proyek perangkat lunak berskala besar [13].

2.1.10 Github

GitHub adalah platform *hosting* repositori Git yang terbesar dan menjadi bagian kolaborasi bagi jutaan pengembang dan proyek. Sebagian besar repositori Git yang ada di dunia di-host di *Github*, menjadikannya alat utama dalam pengembangan perangkat lunak, baik untuk proyek *open-source* maupun pengembangan internal perusahaan. *Github* menyediakan berbagai fitur seperti manajemen repositori, pelacakan masalah (*issue tracking*), tinjauan kode (*code review*), dan otomatisasi alur kerja dengan *CI/CD*. untuk berkolaborasi dalam proyek dengan fitur seperti *pull request*, *branch*, dan *issues*, *GitHub* memungkinkan tim untuk mengusulkan perubahan, melakukan *review*, dan mengintegrasikan perubahan ke dalam kode utama secara terstruktur [13].

2.1.11 XAMPP

XAMPP adalah perangkat lunak yang dirancang untuk mempermudah pengembangan dan pengujian aplikasi berbasis web di lingkungan lokal. Nama XAMPP merupakan singkatan dari beberapa komponen utama yang disertakan di dalamnya, yaitu *X* (*cross-platform*), *Apache* (*web server*), *MySQL/MariaDB*

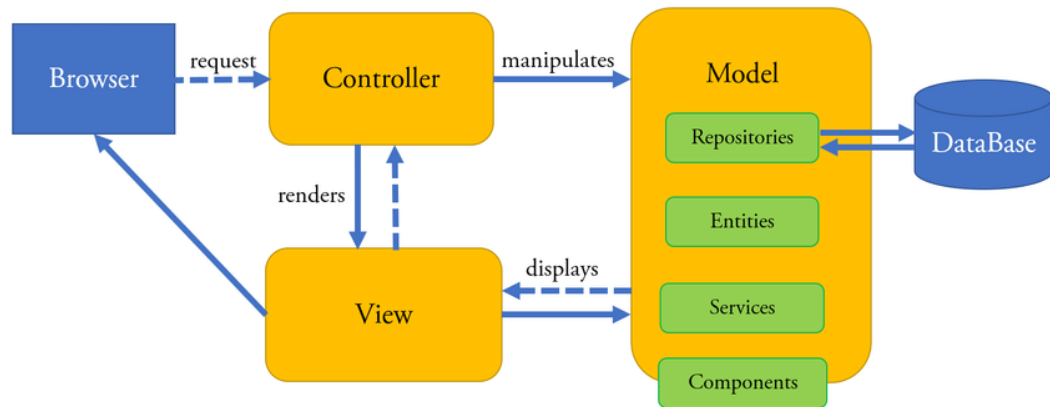
(sistem manajemen basis data), PHP (bahasa pemrograman), dan Perl (bahasa pemrograman tambahan). XAMPP dirancang untuk memudahkan pengguna untuk instalasi serta konfigurasi *server* lokal di dalam perangkat untuk mendukung pengembangan *website* tanpa memerlukan instalasi komponen secara terpisah [14].

2.1.12 Database MySQL

Database adalah kumpulan informasi yang terorganisir dan saling berhubungan, yang dikelola oleh sistem manajemen basis data (DBMS). Terdapat beberapa jenis *database*, termasuk *database* relasional yang berbasis tabel dengan hubungan antar data melalui *foreign key*, *database* objek yang menyimpan data sebagai objek dengan atribut dan perilaku, *database* digital diatur menggunakan *Database Management System (DBMS)* yang menyimpan isi *database* serta mengizinkan pembuatan dan akses pencarian data. Beberapa *database* antara lain : MySQL, SQL Server, MS.Access, Oracle, and PostgreSQL [15].

2.1.13 Laravel

Laravel adalah *framework* PHP yang dirancang untuk membangun aplikasi web modern dengan *syntaks* yang ekspresif dan elegan. *Framework* ini mengadopsi konsep *Model-View-Controller (MVC)* untuk memisahkan logika bisnis dari tampilan, sehingga mempermudah pengelolaan kode. *Laravel* menyediakan fitur-fitur seperti *routing* yang intuitif, sistem autentikasi bawaan, *ORM Eloquent* untuk manajemen basis data, serta alat bantu seperti *Laravel Artisan* untuk otomatisasi tugas pengembangan. Dengan komunitas yang aktif dan dokumentasi yang lengkap, *Laravel* menjadi pilihan populer bagi pengembang yang ingin membangun aplikasi web yang *scalable*, efisien, dan mudah dikembangkan [16].



Gambar 2. 2 Arsitektur Kerangka Kerja Laravel [17].

2.1.14 PHP(*Hypertext Preprocessor*)

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman sisi server yang digunakan secara luas untuk mengembangkan aplikasi web dinamis. PHP memiliki sintaks yang fleksibel dan mudah dipelajari, serta mendukung berbagai *database* seperti MySQL, *PostgreSQL*, dan *SQLite*. Dengan dukungan komunitas yang besar dan ekosistem yang berkembang, PHP telah menjadi bahasa utama dalam pengembangan web modern, terutama dengan kehadiran *framework* menyederhanakan proses pengembangan aplikasi berbasis PHP [16].

2.1.15 *System Usability Scale*

System Usability Scale (SUS) adalah metode evaluasi yang banyak digunakan untuk mengukur tingkat kegunaan suatu sistem atau produk berbasis teknologi melalui kuesioner standar. SUS terdiri dari 10 pertanyaan dengan skala 5 poin yang mencakup aspek kemudahan penggunaan, kompleksitas, serta kepercayaan pengguna terhadap sistem. Perhitungan skala SUS dilakukan dengan mengonversi jawaban menjadi skala numerik, menjumlahkan hasilnya, dan mengalikannya dengan 2,5 untuk mendapatkan nilai dalam rentang 0 hingga 100. Skala SUS diinterpretasikan berdasarkan standar tertentu, di mana nilai di atas 68 dianggap sebagai tingkat *usability* yang baik [18].

2.2 Penelitian Terkait

2.2.1 Evolving Trends in Open-Source RDBMS: Performance, Scalability, and Security Insights

Penelitian ini dilakukan oleh Kumar Miryala pada tahun 2024. Hasil dari penelitian ini membahas perkembangan terbaru dalam sistem manajemen basis data relasional (RDBMS) *open-source*, terutama *MySQL*, *PostgreSQL*, dan *MariaDB*. Penelitian ini bertujuan untuk menganalisis tren, tantangan, dan arah masa depan *open-source* RDBMS dengan fokus pada optimasi kinerja, skalabilitas, dan keamanan. *MySQL* disebut sebagai *database* yang andal dengan performa tinggi, skalabilitas yang baik, serta dukungan komunitas yang luas untuk meningkatkan eksekusi *query*, penyimpanan data, serta perlindungan terhadap ancaman keamanan siber. Dengan semakin berkembangnya teknologi, jurnal ini menyoroti pentingnya inovasi dalam RDBMS *open-source* agar dapat memenuhi kebutuhan manajemen data yang semakin kompleks di masa depan [19].

2.2.2 Design and Implementation of Inventory Management System for University

Penelitian ini dilakukan oleh antosh Soni, Pankaj Chandra, ksha Gupta, Deepak Kant Netam, Sushant Kumar, Kaushik Tiwary pada tahun 2023. Tujuan penelitian ini adalah pengembangan sebuah aplikasi manajemen inventaris berbasis web untuk lingkungan universitas dengan menggunakan teknologi PHP, HTML, CSS, JavaScript, serta *database* SQL. Metode yang digunakan adalah model pengembangan perangkat lunak *Waterfall*, yang mencakup tahap analisis, desain, implementasi, pengujian, dan pemeliharaan. Hasil penelitian menunjukkan bahwa sistem ini mampu mengotomatisasi pencatatan stok, mengurangi kesalahan manusia dalam pembaruan inventaris [20]

2.2.3 Black box testing with Equivalence Partitions Techniques in Transcrop Applications

Penelitian ini dilakukan oleh J Ginanjar Wiro Sasmito, Mirza Alim Mutasodirin, pada tahun 2023. Adapun tujuan dari penelitian ini adalah pengujian sistem

informasi menggunakan *black box testing*, hasil penelitian mengenai *Black box testing* menunjukkan bahwa metode ini efektif dalam mendeteksi kesalahan perangkat lunak tanpa perlu memahami struktur internal kode. Penelitian ini, pengujian dilakukan pada aplikasi *Transcrop* menggunakan teknik *Equivalence Partitioning*, yang membagi data uji ke dalam kelompok yang setara untuk mengidentifikasi validitas input *dan output*. Pengujian dilakukan terhadap 13 kasus uji pada berbagai modul aplikasi, menghasilkan tingkat keberhasilan 77% dan ketidaksesuaian 23%. Hasil ini menunjukkan bahwa teknik *Equivalence Partitioning* secara optimal menemukan ketidaksesuaian fungsionalitas dalam skenario pengujian yang telah ditentukan [21].

2.2.4 Applying agile practices to avoid chaos in User Acceptance Testing: A case study

Penelitian ini dilakukan oleh K.V. Jeeva Padmini, Indika Perera, and H. M. N. Dilum Bandara, pada tahun 2019. Tujuan dari penelitian di atas adalah untuk meningkatkan produktivitas, efisiensi, dan waktu ke pasar (*time to market*) dalam proses *User Acceptance Testing* (UAT) dengan menerapkan praktik *Agile Scrum Framework*. Metode pengembangan yang digunakan pada penelitian ini adalah *Agile scrum framework*, hasil penelitian mengenai *User Acceptance Testing* (UAT) menunjukkan bahwa metode ini berperan penting dalam memastikan bahwa sistem perangkat lunak memenuhi kebutuhan bisnis yang telah disepakati sebelum implementasi penuh, dalam UAT dapat meningkatkan efektivitas pengujian, mengurangi risiko kegagalan proyek, serta meningkatkan kepercayaan pemangku kepentingan terhadap kualitas sistem yang dikembangkan [22].

2.2.5 Implementation of Rapid Application Development Method in the Development of Geographic Information Systems of Industrial Centers

Penelitian ini dilakukan oleh Ginanjar Wiro Sasmito , Dega Surono Wibowo, and Dairoh, pada tahun 2020. Tujuan dari penelitian ini adalah untuk meningkatkan aksesibilitas informasi industri, mempermudah pengumpulan data perusahaan. Adapun metode pengembangan yang digunakan adalah metodologi RAD

melibatkan siklus iteratif yang mencakup perencanaan, analisis, desain, implementasi, dan pengujian dengan fokus pada prototipe yang cepat dan umpan balik dari pengguna. Hasil penelitian menunjukkan bahwa GIS yang dikembangkan dengan RAD dapat diselesaikan dalam waktu 12 minggu, lebih cepat dibandingkan metode *Waterfall*, dengan hasil yang lebih baik dalam hal efisiensi biaya dan optimasi sumber daya [23].

2.2.6 Pengujian Fungsionalitas Sistem Inventaris Barang Pada Sekolah Menengah Kejuruan Citra Borneo Menggunakan *Black box testing*

Penelitian ini dilakukan oleh Fauzan Asrin, pada tahun 2023. Penelitian ini bertujuan untuk mempermudah proses pengelolaan barang, baik bagi pegawai maupun pengelola dari Sekolah Menengah Kejuruan (SMK) Citra Borneo. Selain itu, sistem ini bertujuan untuk meningkatkan efisiensi dan kualitas pengelolaan barang yang diperlukan oleh pegawai atau satuan kerja (satker) di berbagai kawasan Sekolah Menengah Kejuruan (SMK) Citra Borneo. Adapun metode pengujian yang digunakan pada penelitian kali ini ialah *black box testing* yang dimana dari pengujian ini bertujuan untuk meningkatkan efisiensi dan efektivitas dari pengelolaan data inventaris, proses pengujian ini dibagi menjadi tiga tahap yaitu yang pertama perencanaan pengujian, pengembangan kasus pengujian, serta eksekusi pengujian. Berdasarkan hasil pengujian metode *Black Box*, sistem berjalan dengan baik dan memenuhi kriteria yang telah ditetapkan sesuai dengan kebutuhan pengguna, dari hasil pengujian dapat memberikan solusi praktis dan efisien mengenai pengelolaan inventaris di dalam SMK Citra Borneo [24].

2.2.7 Analisis Perbandingan Unjuk Kerja *Database SQL* dan *Database NoSQL* Untuk Mendukung Era *Big Data*

Penelitian ini dilakukan oleh Ari Fadli, Mulki Indana Zulfa, Azis Wisnu Widhi Nugraha, Acep Taryana, dan Muhammad Syaiful Aliim. pada tahun 2020. Penelitian ini bertujuan menganalisis perbandingan kinerja antara *database MySQL* (SQL) dan *Database Redis* (NoSQL) dalam menangani pertumbuhan data di era big data. Metodenya meliputi pengujian waktu eksekusi operasi dasar CRUD (*Create, Read, Update, Delete*) menggunakan PHP pada variasi jumlah data [25]

2.2.8 Sistem Manajemen Pelayanan Pelanggan Menggunakan PHP Dan MySQL (Studi Kasus pada Toko Surya)

Penelitian ini dilakukan oleh Sudaria¹, Arman Syah Putra, Yosua Novembrianto, pada tahun 2021. Penelitian ini bertujuan untuk manajemen pelayanan pelanggan pemasaran dan penjualan secara *online* yang cepat dan akurat supaya memudahkan proses transaksi. Selain itu, sistem ini bertujuan untuk sebagai monitoring manajemen pelayanan pada toko Surya. Adapun metode pengembangan yang digunakan adalah metodologi *Waterfall*. Dalam pengerjaan sistem informasi manajemen pelayanan toko ini menggunakan *XAMPP* sebagai *server side* yang dimana laragon sendiri mendukung berbagai sistem informasi, menyediakan layanan dan fitur seperti, *Apache*, *PHP server*, *PHPMyadmin*, *MySQL*, dan lainnya. *XAMPP* dirancang untuk memudahkan pengguna dalam menginstal dan mengkonfigurasi *server* lokal untuk pengembangan web tanpa perlu *instal* setiap komponennya secara terpisah. Berdasarkan jurnal yang sama, *XAMPP* digunakan dalam pengembangan sistem manajemen pelanggan berbasis web di Toko Surya untuk menjalankan aplikasi berbasis *PHP* di *server* lokal sebelum diterapkan ke *server* produksi. Dengan kemudahan instalasi dan penggunaannya, *XAMPP* menjadi solusi praktis bagi pengembang dalam membangun dan menguji aplikasi berbasis web [26].

2.2.9 Rancang Bangun Aplikasi Web Sekolah Menggunakan UML (*Unified Modeling Language*) Dan Bahasa Pemrograman PHP (*Php Hypertext Preprocessor*)

Penelitian ini dilakukan oleh Abdul Mubarak, pada tahun 2019. Penelitian ini bertujuan membuat *website* Sekolah Menengah Kejuruan (SMK) Plus Indtim. Selain itu, sistem ini bertujuan untuk sebagai penyebaran informasi yang dilakukan oleh pihak sekolah yang sebelumnya menggunakan pamflet dan brosur. Adapun metode pengembangan yang digunakan adalah UML (*Unified Modeling language*). Dalam pengerjaan sistem informasi manajemen proyek ini menggunakan UML sebagai visualisasi sistem desain agar lebih mudah, pada pengembangan memilih *UML* dikaren dari pihak pengembang dapat mudah memvisualisasikan segala aspek yang direncan dalam pembuatan *website*, termasuk struktur, interaksi dan alur kerja,

yang membantu dalam memahami dan mengkomunikasikan desain sistem secara lebih jelas. Dalam penelitian kali ini penggunaan UML berupa diagram seperti *use case*, *Activity*, *sequence* dan *class diagram*. Yang berfungsi sebagai panduan dalam proses pengembangan, penggunaan UML juga bukan hanya untuk meningkatkan pemahaman tim pengembangan, tetapi juga agar memastikan bahwa semua kebutuhan pengguna dapat diakomodasi dengan baik, sehingga menghasilkan aplikasi yang lebih terstruktur dan sesuai dengan harapan pengguna [27].

2.2.10 Evaluasi Kegunaan Aplikasi Akhlaqu Menggunakan Metode *System Usability Scale* (SUS) dan Implementasi *Indexing* pada MongoDB

Penelitian ini dilakukan oleh Mochammad Fauzan Fadilah, Nining Rahaningsih, Raditya Danar Dana pada tahun 2024. Penelitian ini bertujuan Mengevaluasi tingkat kegunaan Aplikasi Akhlaqu yang digunakan di Sekolah Dasar Islam Terpadu Al-Falah dengan metode *System Usability Scale* (SUS) serta menganalisis dampak penerapan teknik *indexing MongoDB* terhadap performa aplikasi. Penelitian ini menggunakan metode kuantitatif dengan penyebaran kuesioner kepada pengguna aplikasi. Data dianalisis menggunakan *System Usability Scale* (SUS). Selain itu, dilakukan uji validitas, reliabilitas. Hasil penelitian menunjukkan bahwa Aplikasi Akhlaqu memiliki tingkat usability yang tinggi dengan skala SUS sebesar 89,7 (grade B), yang menandakan bahwa aplikasi tersebut mudah digunakan dan memenuhi standar *usability* yang baik

III. METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

Pelaksanaan penelitian dan pengembangan *website* ini dilakukan pada waktu dan tempat yang telah ditentukan sebagai berikut:

Waktu : Februari 2025 – Juli 2025

Tempat : Unit Pelayanan Kesehatan DPR-RI dan Laboratorium komputer Teknik Informatika Universitas Lampung

Tabel 3. 1 Waktu Penelitian

| No. | Aktivitas | Feb | | | | Mar | | | | Apr | | | | Mei | | | | Jun | | | | Jul | | | |
|-----|---------------------|-----|---|---|---|-----|---|---|---|-----|---|---|---|-----|---|---|---|-----|---|---|---|-----|---|---|---|
| | | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 1. | Analisis kebutuhan | | | | | | | | | | | | | | | | | | | | | | | | |
| 2. | Desain interface | | | | | | | | | | | | | | | | | | | | | | | | |
| 3. | Implementasi sistem | | | | | | | | | | | | | | | | | | | | | | | | |
| 4. | Pengujian | | | | | | | | | | | | | | | | | | | | | | | | |
| 5. | Evaluasi sistem | | | | | | | | | | | | | | | | | | | | | | | | |
| 6. | Penulisan laporan | | | | | | | | | | | | | | | | | | | | | | | | |

3.2 Alat Penelitian

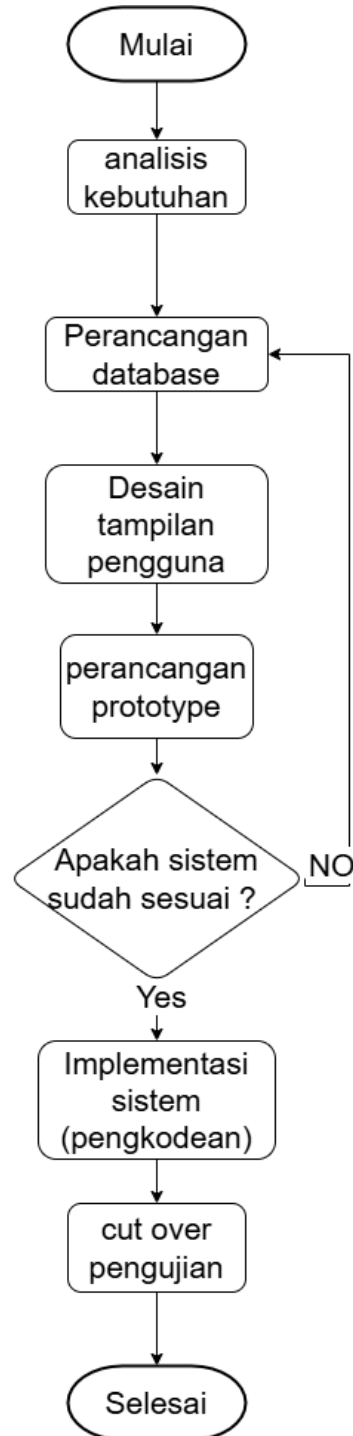
Adapun alat yang digunakan dalam penelitian ini adalah sebagai berikut:

Tabel 3. 2 Alat dan bahan

| No | Nama | Spesifikasi | Deskripsi |
|----|---------------------------|--|---|
| 1. | Laptop | <i>Prosesor AMD Ryzen 7, RAM 16 GB, ROM 512 GB, Windows 11 64bit</i> | Perangkat keras yang digunakan untuk pengembangan aplikasi dan penulisan laporan penelitian. |
| 2. | <i>Figma</i> | Versi 124.2.9 | Perangkat lunak untuk membuat dan melihat desain |
| 3. | <i>Visual Studio Code</i> | Versi 1.93 | Perangkat lunak editor kode yang digunakan untuk pengembangan aplikasi. |
| 4. | <i>PHP</i> | 8.2.12 | Bahasa pemrograman untuk pengembangan <i>Website</i> |
| 5. | <i>Laravel</i> | Versi 11.44.0 | <i>Framework</i> untuk pengembangan <i>website</i> yang memungkinkan pembuatan <i>website</i> menggunakan bahasa pemrograman <i>PHP</i> |
| 6. | <i>XAMPP</i> | Versi 3.3.0 | Aplikasi <i>server side</i> menyimpan <i>database</i> . |
| 7. | Data arsip barang & surat | - | Data dari arsip barang, surat masuk dan surat keluar dari Yankes DPR-RI |

3.3 Tahapan Penelitian

Penelitian ini menggunakan metodologi *Rapid Application Development* dengan alur seperti pada Gambar 3.1



Gambar 3. 1 Tahapan Penelitian

3.3.1 Analisis Kebutuhan

Analisis kebutuhan adalah tahapan yang dilakukan pertama pada penelitian. Tahap ini bertujuan untuk menghasilkan daftar kebutuhan pengguna, kebutuhan fungsional dan non-fungsional dari sistem yang dikembangkan. Kebutuhan tersebut diperoleh melalui hasil wawancara dan observasi dari Bagian Layanan Kesehatan DPR-RI(staf/kepala).

Tabel 3. 3 Borang Pertanyaan fungsionalitas dan non fungsional sistem

| No | Pertanyaan | Jawaban |
|----|---|--|
| 1. | Bagaimana kondisi pencatatan inventaris dan data administrasi saat ini pada bagian layanan kesehatan DPR-RI | Kondisi saat ini mengenai pencatatan barang dan arsip surat, masih dilakukan secara manual melalui excel maupun buku |
| 2. | Apakah dengan adanya pembuatan sistem informasi inventaris dan data administrasi dapat membantu bagian pelayanan DPR-RI | Dengan adanya digitalisasi pencatatan barang dan pengajuan cuti, dapat membantu pelayanan di Yankes DPR-RI |
| 3. | Seberapa penting untuk digitalisasi sebuah inventaris barang dan data administrasi di Yankes DPR-RI | Digitalisasi diperlukan agar mengoptimalkan pencatatan barang dan melihat pengajuan cuti menjadi lebih baik |
| 4. | Bagaimana kondisi disana saat kepala bagian membutuhkan laporan terkait barang ataupun surat ? | Hasil pelaporan masih melalui <i>Excel</i> , <i>pdf</i> dikirim melalui <i>Whatsapp</i> |
| 5. | Apakah diperlukan laporan berdasarkan kategori tertentu (misal: laporan berdasarkan jenis barang, laporan barang, maupun arsip surat) | Yang dibutuhkan dalam laporan, seperti laporan barang masuk, laporan barang keluar, dan laporan stok barang |
| 6. | Pada proses pencatatan administrasi/inventaris barang yang terjadi di sana apakah terdapat alur | Proses pengajuan/input data dilakukan oleh karyawan (admin) disalurkan kepada |

| No | Pertanyaan | Jawaban |
|----|---|--|
| | menerima ataupun memberikan laporan terkait inventaris barang/data administrasi | kepala unit untuk validasi atau penerimaan |
| 7. | Bagaimana proses pencarian yang terjadi jika kepala unit membutuhkan laporan barang/arsip data administrasi ? | Proses pencarian dilakukan dengan mencari tanggal barang/arsip data administrasi tersebut di dalam <i>Excel</i> maupun berdasarkan tanggal pengajuan pada pdf tersebut |

3.3.1.1 Kebutuhan Pengguna, Kebutuhan Fungsional, dan Kebutuhan Non Fungsional

Berdasarkan hasil dari wawancara kepada staf dan observasi langsung di dalam Yankes DPR-RI. Diperoleh kebutuhan fungsional dan non fungsionalitas pengguna. Adapun kebutuhan pengguna dari sistem ini adalah sebagai berikut:

1. Pengguna ingin menambahkan daftar barang di dalam *dashboard*.
2. Pengguna ingin menghapus daftar barang di dalam *dashboard*.
3. Pengguna ingin mengedit daftar barang di dalam *dashboard*.
4. Pengguna ingin menambahkan daftar peminjaman barang
5. Pengguna ingin menambahkan akun *employee*.
6. Pengguna ingin melihat daftar barang dan data administrasi di dalam *dashboard*.
7. Pengguna ingin melihat laporan barang masuk/keluar dan *stock* barang
8. Pengguna ingin mengelola data profil pribadi.
9. Pengguna ingin cetak laporan barang masuk/ keluar
10. Pengguna ingin cetak laporan *stock* barang

Kebutuhan fungsional menggambarkan fitur-fitur yang harus ada dalam sistem untuk memastikan *website* dapat berfungsi dengan baik sesuai dengan kebutuhan

pengguna yang telah ditetapkan. Berikut merupakan tabel kebutuhan fungsional pada perancangan sistem ini:

Tabel 3. 4 Kebutuhan fungsional

| ID | Penjelasan |
|-------|---|
| KF-01 | Sistem dapat menyediakan layanan bagi pengguna untuk masuk dan keluar sistem |
| KF-02 | Sistem dapat menyediakan layanan bagi pengguna untuk menambahkan <i>Employee</i> (admin kantor) |
| KF-03 | Sistem dapat menambahkan <i>list</i> barang dan data administrasi kantor di dalam <i>dashboard</i> |
| KF-04 | Sistem dapat menghapus <i>list</i> barang dan data administrasi kantor di dalam <i>dashboard</i> |
| KF-05 | Sistem dapat mengedit daftar barang dan data administrasi kantor di dalam <i>dashboard</i> |
| KF-06 | Sistem dapat melihat <i>list</i> barang dan data administrasi kantor di dalam <i>dashboard</i> |
| KF-07 | Sistem memiliki fitur cetak laporan yang memungkinkan pengguna untuk membuat laporan dalam format excel. |
| KF-08 | Sistem memiliki fitur untuk memfilter jenis barang ataupun data administrasi kantor pada saat mencari kata. |
| KF-09 | Sistem dapat menyediakan layanan bagi pengguna untuk menambahkan barang ataupun data administrasi kantor |
| KF-10 | Sistem dapat menyediakan layanan bagi pengguna untuk mengubah foto dan data profil |
| KF-11 | Sistem dapat membuat kategori barang |

Kebutuhan non fungsional mencakup aspek-aspek teknis yang mendukung kualitas, performa, dan keamanan aplikasi. Berikut merupakan tabel kebutuhan *non* fungsional pada perancangan sistem ini:

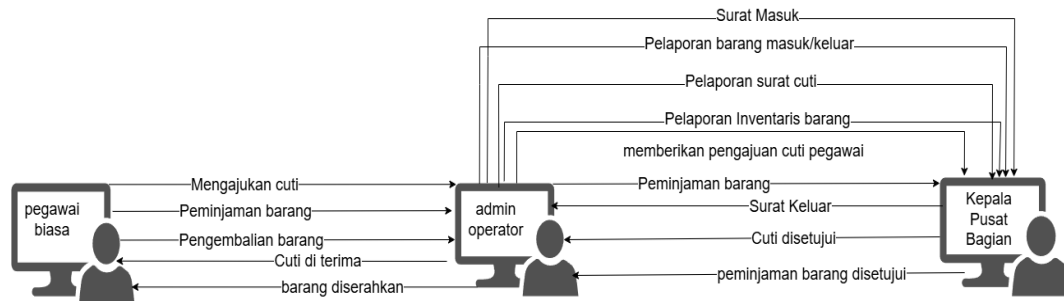
Tabel 3. 5 Kebutuhan non fungsional

| ID | Parameter | Penjelasan |
|--------|-------------------------------|---|
| KnF-01 | <i>Availability</i> | <ul style="list-style-type: none"> Aplikasi harus selalu dapat diakses sepanjang waktu |
| KnF-02 | <i>Reliability</i> | <ul style="list-style-type: none"> Sistem harus mampu menangani kesalahan input atau kegagalan eksternal dengan memberikan pesan yang informatif kepada pengguna tanpa menyebabkan <i>crash</i> |
| KnF-03 | <i>Portability</i> | <ul style="list-style-type: none"> Sistem harus dapat diakses dan berfungsi dengan baik melalui berbagai <i>browser</i> populer (<i>Chrome, Firefox, Safari, Edge</i>) melalui desktop atau <i>mobile</i> |
| KnF-04 | <i>Performance Efficiency</i> | <ul style="list-style-type: none"> Sistem harus memiliki waktu respons kurang dari 5 detik untuk semua proses utama, termasuk pencarian dan pemuatan halaman. |
| KnF-05 | <i>security</i> | <ul style="list-style-type: none"> Sistem memiliki pembatasan akses berdasarkan role |
| KnF-06 | <i>Usability</i> | <ul style="list-style-type: none"> Antarmuka pengguna harus dirancang agar mudah dipahami dan digunakan secara intuitif. Pengguna harus mampu menyelesaikan tugas utama dalam waktu kurang dari 5 menit melalui langkah-langkah yang mudah dipahami dan efisien |
| KnF-07 | <i>Maintainability</i> | <ul style="list-style-type: none"> Sistem mudah dipelihara dan diperbarui |

3.3.2 Tahapan Iterasi (Perancangan Desain)

Rancangan sistem dijelaskan menggunakan beberapa diagram agar pengguna dapat paham bagaimana alur dari proses berjalannya sistem tersebut.

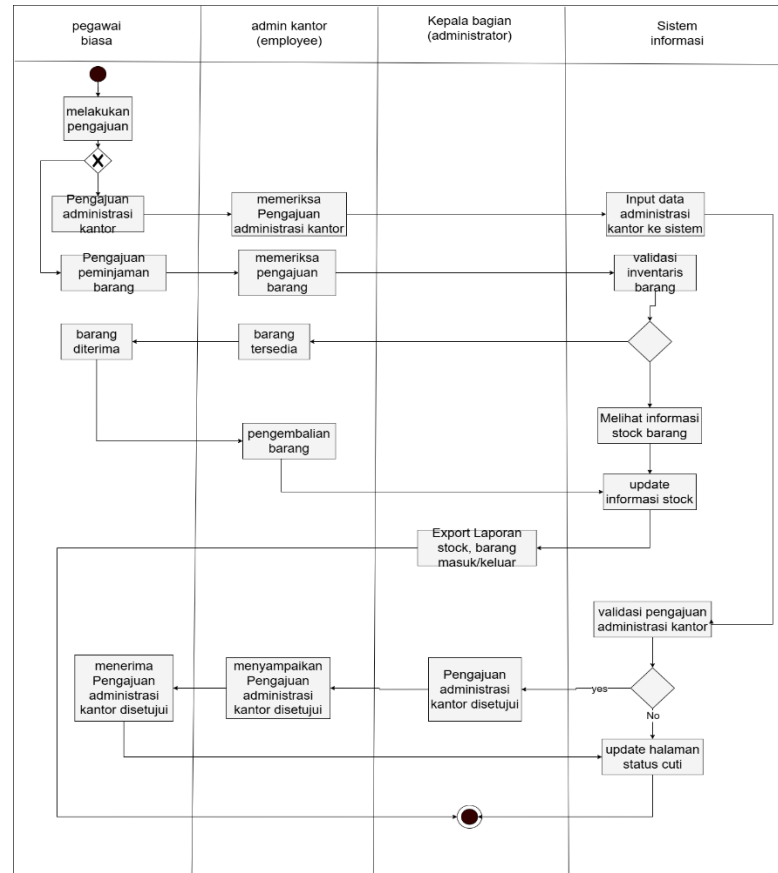
3.3.2.1 *Business process* (Yankes saat ini)



Gambar 3. 2 Business process saat ini

berdasarkan gambar 3.2 di atas yang menggambarkan *business process* yang sedang berjalan saat ini di Yankes. Diagram ini menunjukkan alur kerja mulai dari peran pegawai biasa, admin operator kantor, hingga kepala pusat bagian dalam mengelola berbagai proses seperti peminjaman barang, pengajuan cuti, arsip data administrasi, hingga pembuatan laporan barang.

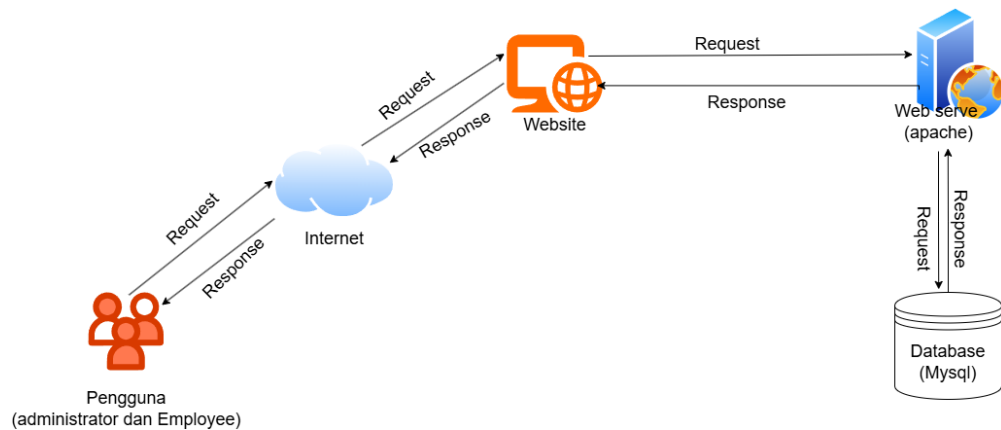
3.3.2.2 *Business process* (Rekomendasi digitalisasi)



Gambar 3. 3 *Business process Recommendation* (digitalisasi)

Proses bisnis pada diagram ini menggambarkan digitalisasi alur pengajuan cuti dan peminjaman barang di lingkungan layanan kesehatan DPR RI (Yankes DPR RI), dengan melibatkan peran pegawai biasa, admin (*employee*), kepala bagian (*administrator*), dan sistem informasi. Pegawai dapat melakukan pengajuan baik cuti maupun peminjaman barang melalui sistem. Pengajuan cuti diperiksa oleh admin, dilanjutkan dengan proses validasi oleh sistem dan persetujuan oleh kepala bagian. Jika disetujui, status cuti diperbarui dan disampaikan kembali ke pegawai. Tahapan peminjaman barang, admin memeriksa ketersediaan dan validasi inventaris dilakukan oleh sistem. Jika barang tersedia, maka diberikan kepada pegawai dan nantinya dikembalikan. Sistem juga memungkinkan *update* data stok barang serta cetak laporan barang masuk dan keluar.

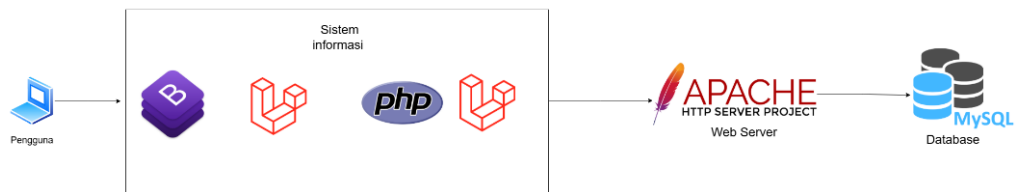
3.3.2.3 Alur sistem



Gambar 3. 4 Alur Sistem

Sistem inventaris pada gambar menunjukkan alur komunikasi data yang dimulai dari Pengguna (*Administrator* dan *Employee*) yang mengirimkan permintaan (*request*) melalui Internet ke *Website* sistem inventaris. *Website* kemudian meneruskan permintaan tersebut ke *Web Server (Apache)* yang bertindak sebagai pengolah utama aplikasi. *Web Server* berkomunikasi dengan *Database (MySQL)* untuk mengambil atau menyimpan data inventaris sesuai kebutuhan. Setelah data diproses, hasilnya dikembalikan melalui jalur yang sama namun dalam arah berlawanan dari *database* ke *Web Server*, lalu ke *Website*, melewati Internet, dan akhirnya sampai ke Pengguna dalam bentuk respons (*response*) yang menampilkan informasi inventaris yang diminta.

3.3.2.4 Arsitektur sistem



Gambar 3. 5 Arsitektur Sistem

Arsitektur sistem pada gambar menunjukkan struktur aplikasi web inventaris dengan pendekatan yang terdiri dari empat komponen utama yang saling terhubung. dimulai dari Pengguna yang berinteraksi dengan *Frontend* (menggunakan *Bootstrap* dan *Laravel* untuk antarmuka), kemudian permintaan diteruskan ke *Backend* (dibangun dengan *PHP* dan *Laravel* sebagai *framework*) *Backend* berkomunikasi dengan *Web Server* (*Apache*) yang menangani permintaan *HTTP* dan mengelola layanan *web*, dan terhubung ke *database* (*MySQL*) yang menyimpan seluruh data inventaris seperti informasi barang, stok, transaksi, dan data pengguna.

3.3.2.5 Use Case

Use case menggambarkan skenario penggunaan sistem terkait bagaimana interaksi antara aktor (pengguna) dengan sistem yang dikembangkan. *Use case* ini bertujuan untuk menggambarkan kebutuhan dan perilaku yang dapat dilakukan oleh pengguna, berdasarkan peran yang dimiliki setiap aktor. Setiap aktor, memiliki peran dan akses yang berbeda terhadap fitur-fitur dalam sistem. Setiap *use case* menunjukkan fitur atau fungsi yang dapat dilakukan oleh pengguna.



Gambar 3.6 Use Case

Berikut merupakan tabel definisi aktor pada *use case*:

Tabel 3. 6 Definisi *use case*

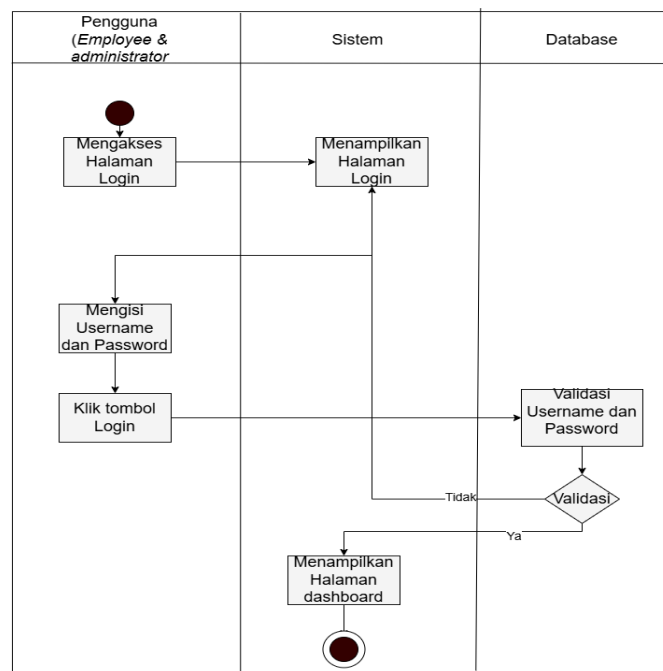
| Aktor | Deskripsi |
|--|---|
| Administrator (Kepala bagian Bagian) | Aktor dengan peran ini dapat mengakses menu <i>dashboard</i> , dan dapat melakukan hal hal yang dilakukan oleh aktor <i>employee</i> (admin kantor) selain itu, aktor ini juga dapat mengakses fitur tambahan, berupa menambahkan <i>employee</i> , |

| | |
|-----------------------------------|---|
| | validasi cuti, cetak laporan barang masuk/keluar, dan stock barang |
| <i>Employee</i> (admin kantor) | Aktor dengan peran ini dapat mengakses menu <i>dashboard</i> . Peran ini dapat melakukan hal-hal meliputi, fitur <i>input</i> , hapus, <i>edit</i> barang dan data administrasi, melihat laporan barang masuk/keluar, menyunting <i>profile</i> , serta melakukan pengajuan cuti. |

3.3.2.6 Activity diagram

Activity diagram adalah penggambaran dari alur kerja atau aktivitas dalam sistem yang digunakan untuk memvisualisasikan proses, tindakan, atau keputusan yang dilakukan pengguna maupun sistem. Diagram ini menunjukkan urutan langkah-langkah dari awal hingga akhir sehingga membantu memahami bagaimana alur aktivitas dalam sistem dikembangkan

a. Activity diagram login

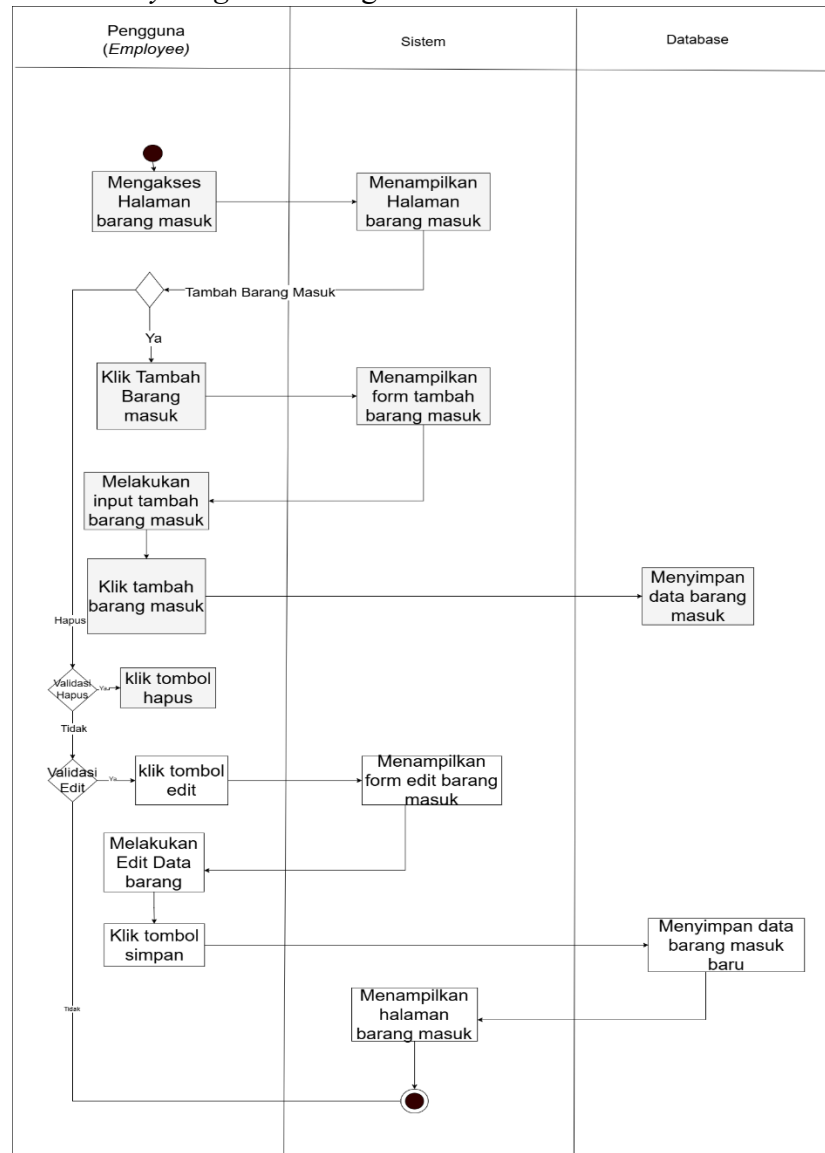


Gambar 3. 7 Activity diagram login

Gambar 3.7 menunjukkan alur *activity diagram* kedalam sistem. *Activity* dimulai dari pengguna mengakses halaman *login* dan mengisi formulir dengan mengisi *username* dan *password*. Setelah itu pengguna klik tombol *login*, Sistem

selanjutnya melanjutkan pengecekan kedalam *database* mengenai *username* dan *password* yang dimasukkan, jika *username* dan *password* salah maka pengguna dikembalikan lagi dalam halaman *login*, jika hasil sesuai maka sistem melanjutkan pengguna dapat masuk ke dalam halaman utama *dashboard*

b. *Activity diagram* barang masuk

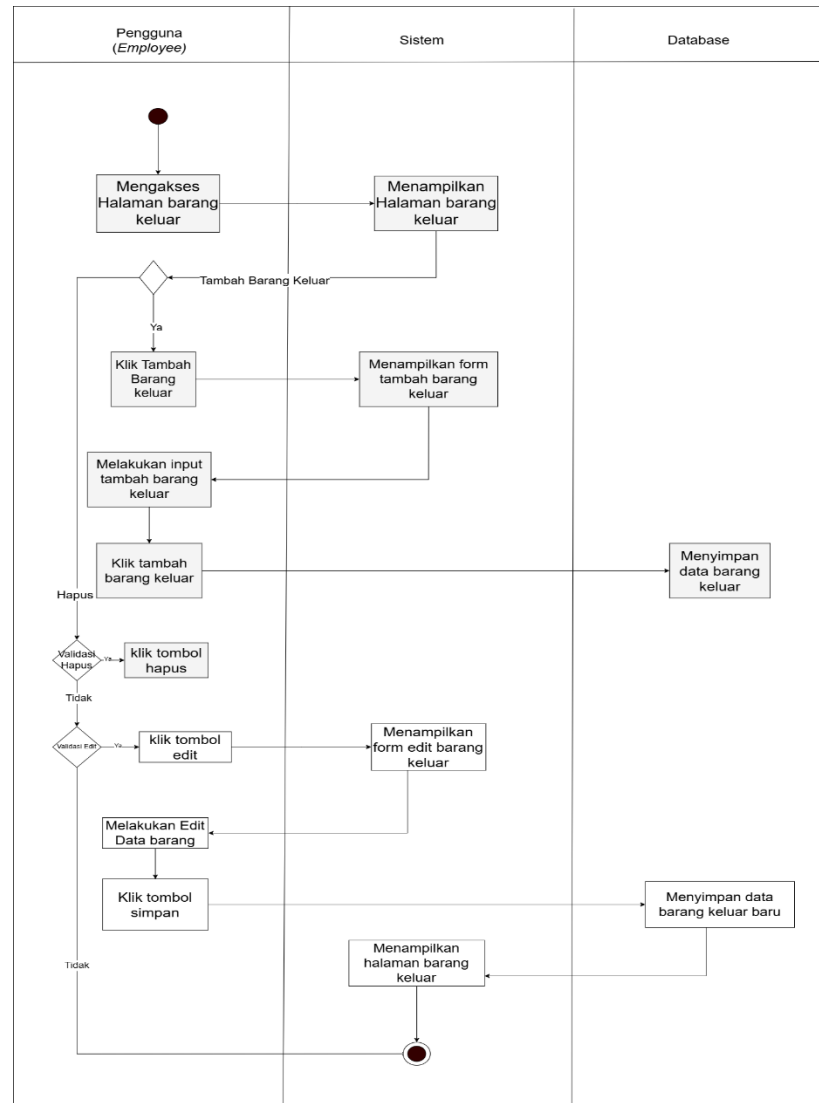


Gambar 3. 8 *activity diagram* barang masuk

gambar 3.8 *Activity diagram* barang masuk yang di mana proses dimulai dari aktor pengguna yang mengakses halaman barang masuk, kemudian sistem menampilkan halaman tersebut. Setelah itu terdapat *decision point* "Tambah Barang Masuk" di mana jika Pengguna memilih "Ya", maka Pengguna klik tombol tambah barang

masuk dan sistem menampilkan formulir tambah barang masuk. Pengguna kemudian melakukan input tambah barang masuk dan klik tombol simpan, yang mengakibatkan data barang masuk disimpan ke *database*. Selain itu, terdapat pilihan untuk hapus atau *edit* data,

c. *Activity diagram* barang keluar



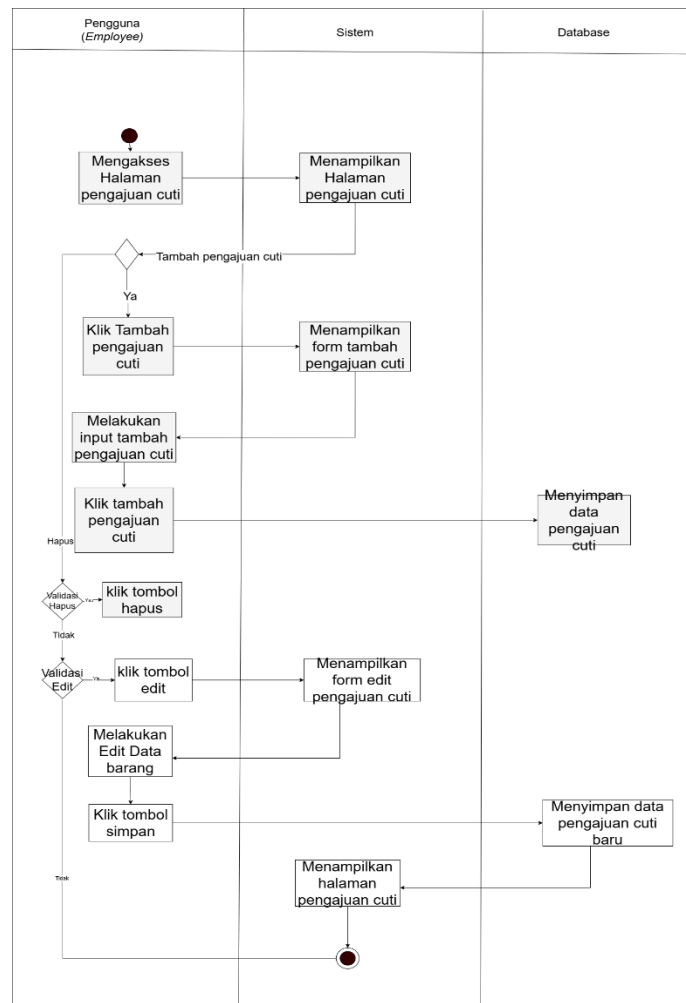
Gambar 3. 9 *Activity diagram* barang keluar

Berdasarkan gambar 3.9 *activity diagram* barang keluar diatas, dimulai dari titik start awal pengguna, alur mengalir ketika pengguna mengakses halaman barang keluar dan sistem merespons dengan menampilkan halaman tersebut. Selanjutnya, pengguna dihadapkan pada pilihan untuk menambah barang keluar yang divisualisasikan melalui *decision node*. Jika pengguna memilih "Ya", proses berlanjut dengan klik tombol tambah, sistem menampilkan *form input*, pengguna

memasukkan data, dan klik tombol simpan yang kemudian direspon oleh *database* dengan menyimpan informasi barang keluar mencapai titik keputusan untuk validasi operasi hapus dan *edit*, di mana pengguna dapat memilih untuk menghapus data atau melanjutkan ke opsi pengeditan.

Pada bagian berikutnya, jika validasi *edit* menghasilkan "Ya", sistem menampilkan *form edit* yang memungkinkan pengguna melakukan modifikasi data dan menyimpannya kembali ke *database*

d. *Activity diagram* pengajuan cuti

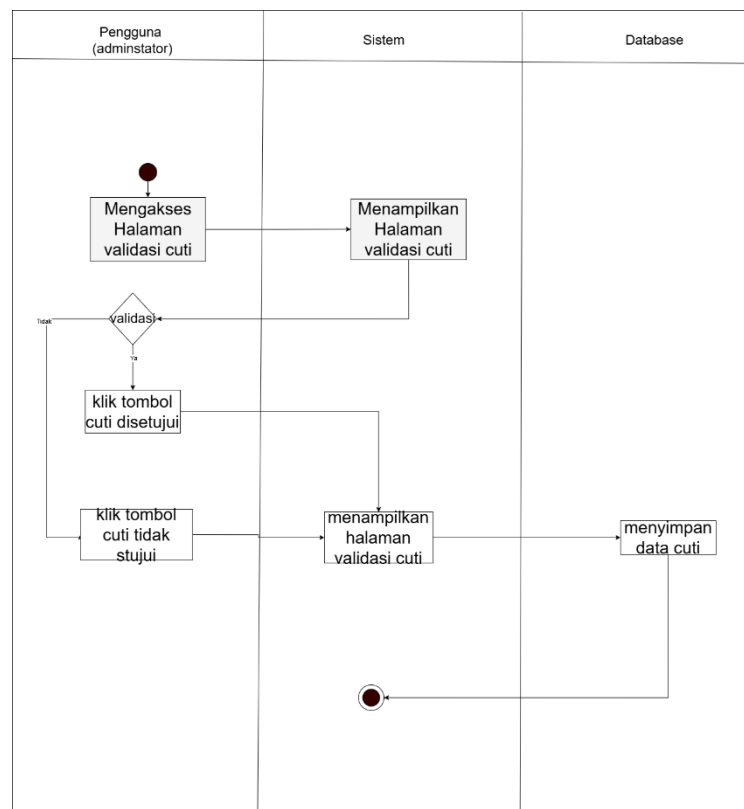


Gambar 3. 10 *Activity diagram* pengajuan cuti

Berdasarkan gambar 3.15 di atas *activity diagram* pengajuan cuti di atas, alur dimulai ketika pengguna mengakses halaman pengajuan cuti, yang kemudian sistem menampilkan halaman tersebut. Jika pengguna ingin menambah pengajuan cuti, mereka memilih opsi "Tambah pengajuan cuti", yang membuat sistem menampilkan *form* untuk menambah pengajuan cuti. Pengguna kemudian mengisi

form tersebut dan klik tombol simpan, yang mengarah pada sistem untuk menyimpan data pengajuan cuti ke dalam *database*. Jika pengguna memilih untuk *mengedit* pengajuan cuti, mereka mengklik tombol *edit*, yang memunculkan *form* untuk mengedit pengajuan cuti yang telah ada. Setelah melakukan perubahan, pengguna menyimpan data yang telah diedit, dan sistem menyimpan data baru tersebut ke dalam *database*. Jika pengguna memutuskan untuk menghapus pengajuan cuti, mereka klik tombol hapus, yang meminta validasi penghapusan. Setelah validasi hapus, sistem menghapus data pengajuan cuti tersebut dari *database*. Setelah seluruh proses selesai, sistem menampilkan kembali halaman pengajuan cuti yang diperbarui.

e. *Activity diagram* validasi cuti

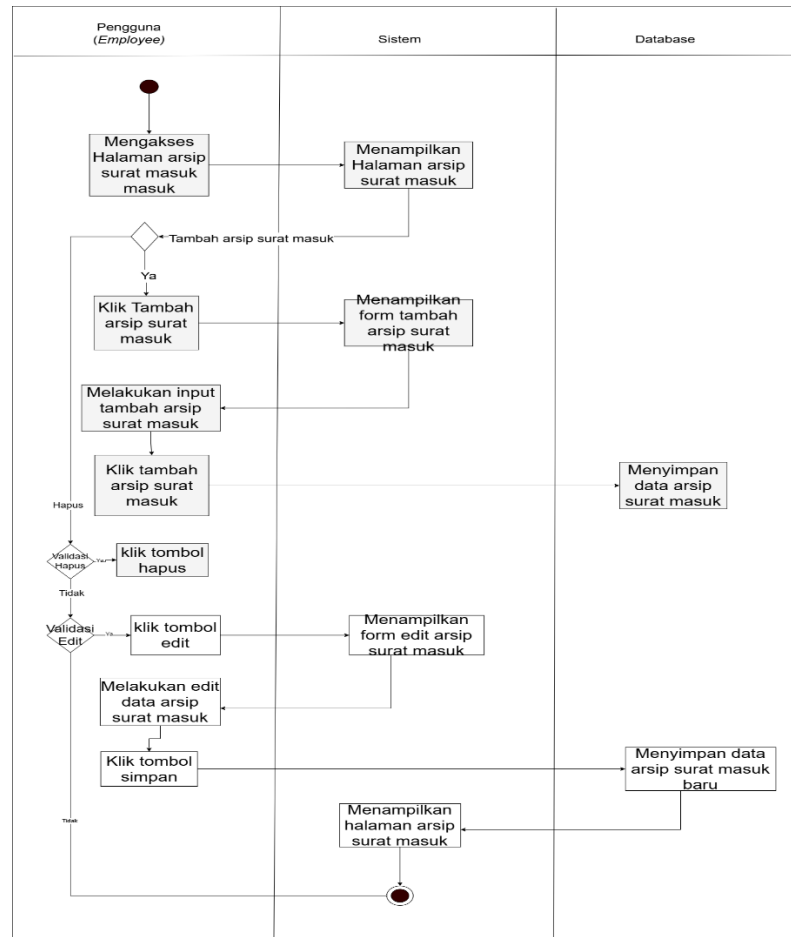


Gambar 3. 11 *Activity diagram* validasi cuti

berdasarkan gambar 3.16 *activity diagram* validasi cuti di atas, alur dimulai ketika kepala unit mengakses halaman validasi cuti yang ditampilkan oleh sistem. Setelah itu, kepala unit melakukan validasi terhadap pengajuan cuti yang ada. Jika kepala unit menyetujui pengajuan cuti, mereka klik tombol "cuti disetujui", yang menyebabkan sistem menyimpan data pengajuan cuti yang telah disetujui ke dalam

database. Sebaliknya, jika kepala unit tidak menyetujui pengajuan cuti, mereka klik tombol "cuti tidak disetujui", yang membuat sistem kembali menampilkan halaman validasi cuti, dan data pengajuan cuti tidak disimpan. Alur ini memastikan bahwa pengajuan cuti yang disetujui oleh kepala unit tercatat dalam sistem dan tersimpan dengan benar dalam *database*.

f. *Activity diagram* arsip surat

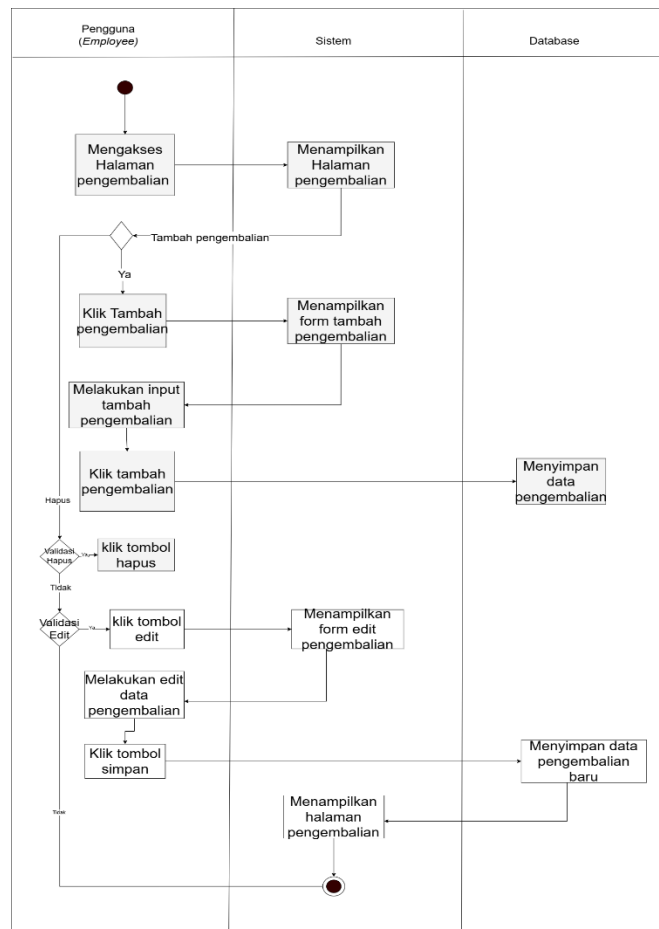


Gambar 3. 12 *Activity diagram* arsip surat

Berdasarkan gambar 3.17 *activity diagram* diatas, alur dimulai ketika pengguna mengakses halaman surat, yang kemudian sistem menampilkan halaman tersebut. Jika pengguna ingin menambah surat, mereka memilih opsi "Tambah surat", yang menyebabkan sistem menampilkan *form* untuk memasukkan data surat baru. Setelah mengisi *form*, pengguna klik tombol "simpan", yang memicu sistem untuk menyimpan data surat yang baru tersebut ke dalam *database*. Jika pengguna memilih untuk mengedit surat, maka klik tombol "edit", yang kemudian menampilkan *form* untuk melakukan perubahan pada data surat yang sudah ada.

Setelah melakukan edit, pengguna menyimpan perubahan tersebut, dan sistem menyimpan data surat yang telah diperbarui ke dalam *database*. Jika pengguna ingin menghapus surat, mereka klik tombol "hapus", dan sistem memvalidasi penghapusan tersebut. Jika validasi penghapusan berhasil, sistem menghapus data surat dari *database*. Setelah semua aksi selesai, sistem kembali menampilkan halaman surat yang diperbarui.

g. *Activity diagram pengembalian*

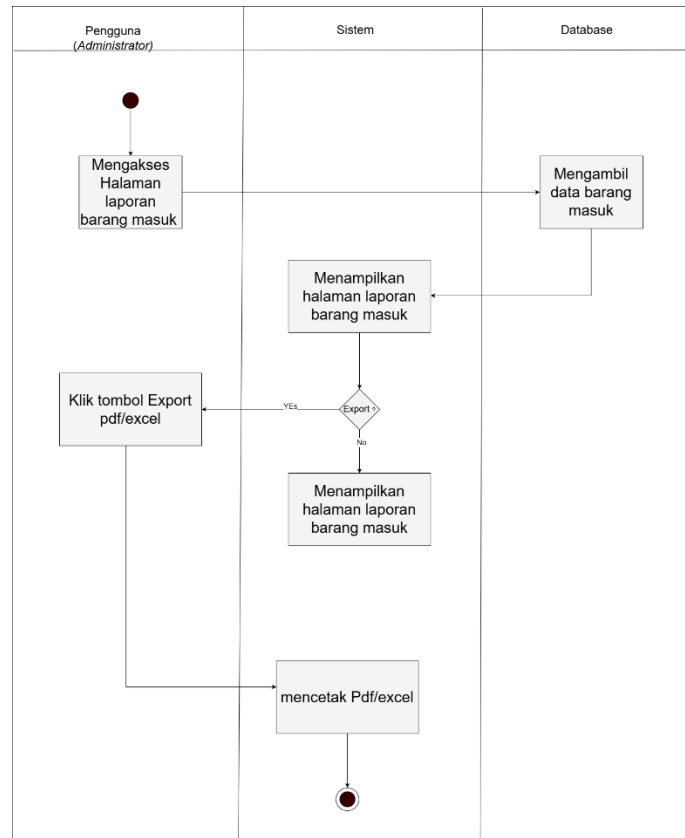


Gambar 3. 13 *activity diagram* pengembalian

Berdasarkan gambar 3.18 di atas, *activity diagram* pengembalian, menggambarkan alur proses pengelolaan pengembalian dalam sistem yang melibatkan tiga aktor utama yaitu Pengguna (*Employee*), Sistem, dan *database*. Proses dimulai saat pengguna mengakses halaman pengembalian yang kemudian ditampilkan oleh sistem. Pengguna memiliki opsi untuk menambah pengembalian dengan klik tombol tambah, yang menghasilkan tampilan *form input* data baru. Setelah melakukan input data dan klik tombol tambah pengembalian, data tersebut

disimpan ke *database*. Selanjutnya terdapat validasi penghapusan data melalui tombol hapus dan modifikasi data melalui tombol *edit* yang memungkinkan pengguna melakukan perubahan pada data pengembalian yang sudah ada. Setiap perubahan pada data disimpan kembali ke *database* dan sistem menampilkan halaman pengembalian yang sudah diperbarui sebagai akhir dari alur proses.

h. *Activity diagram* laporan barang masuk

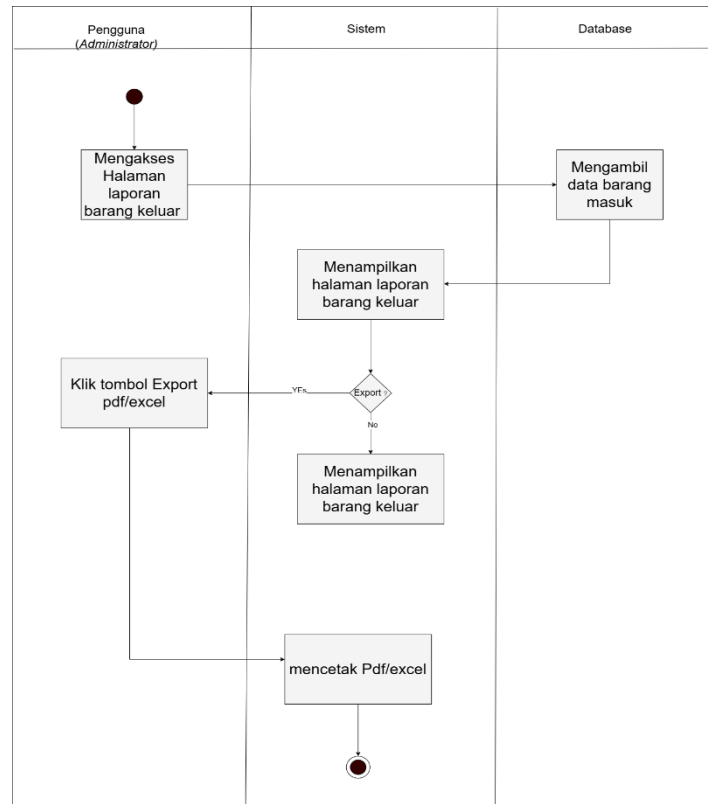


Gambar 3. 14 *activity diagram* laporan barang masuk

Berdasarkan gambar 3.19 di atas, *activity diagram* menggambarkan alur proses pembuatan laporan barang masuk yang melibatkan tiga entitas yaitu Pengguna (Administrator), Sistem, dan *database*. Proses dimulai ketika Administrator mengakses halaman laporan barang masuk, yang kemudian memicu pengambilan data barang masuk dari *database*. Data yang diambil kemudian digunakan sistem untuk menampilkan halaman laporan barang masuk. Setelah halaman ditampilkan, terdapat *decision point* apakah melakukan cetak laporan atau tidak; jika Ya, Administrator klik tombol Cetak PDF/Excel yang dilanjutkan dengan sistem mencetak laporan dalam format PDF atau Excel, namun jika tidak melakukan cetak, sistem tetap menampilkan halaman laporan barang masuk. Alur proses berakhir

setelah laporan berhasil dicetak atau admin memilih untuk tidak melakukan cetak laporan. Diagram ini menunjukkan bahwa fungsi utama dari halaman laporan barang masuk adalah untuk menampilkan data dan menyediakan opsi untuk mengekspor data dalam format yang dapat dicetak.

i. *Activity diagram* laporan barang keluar

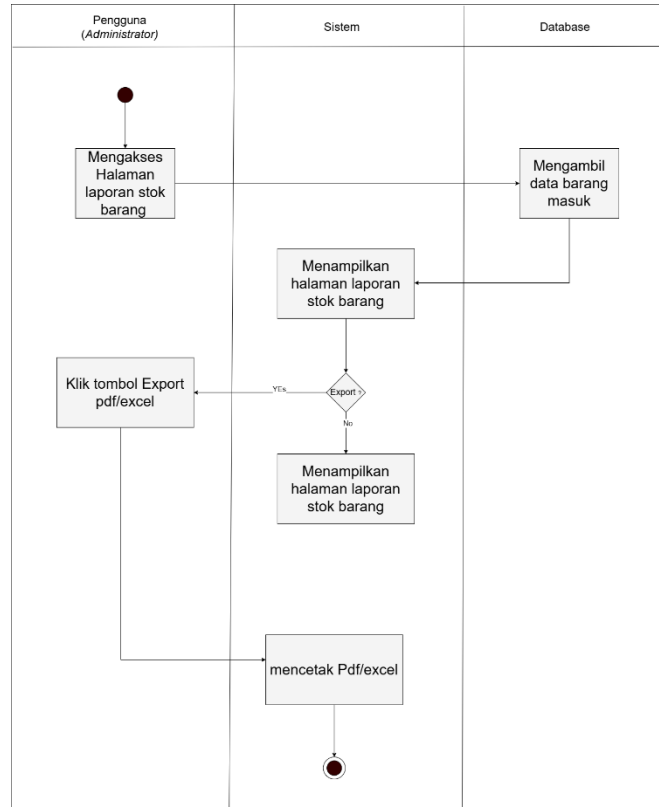


Gambar 3. 15 *activity diagram* laporan barang masuk

Berdasarkan gambar 3.20 di atas, *activity diagram* menggambarkan alur proses pembuatan laporan barang keluar yang melibatkan tiga entitas yaitu Pengguna (Administrator), Sistem, dan *database*. Proses dimulai ketika Administrator mengakses halaman laporan barang keluar, yang kemudian memicu pengambilan data barang keluar dari *database*. Data yang diambil kemudian digunakan sistem untuk menampilkan halaman laporan barang keluar. Setelah halaman ditampilkan, terdapat *decision point* apakah melakukan cetak laporan atau tidak; jika Ya, Administrator klik tombol *Cetak PDF/Excel* yang dilanjutkan dengan sistem mencetak laporan dalam format *PDF* atau Excel, namun jika tidak melakukan *cetak*, sistem tetap menampilkan halaman laporan barang keluar. Alur proses

berakhir setelah laporan berhasil dicetak atau Administrator memilih untuk tidak melakukan *cetak* laporan.

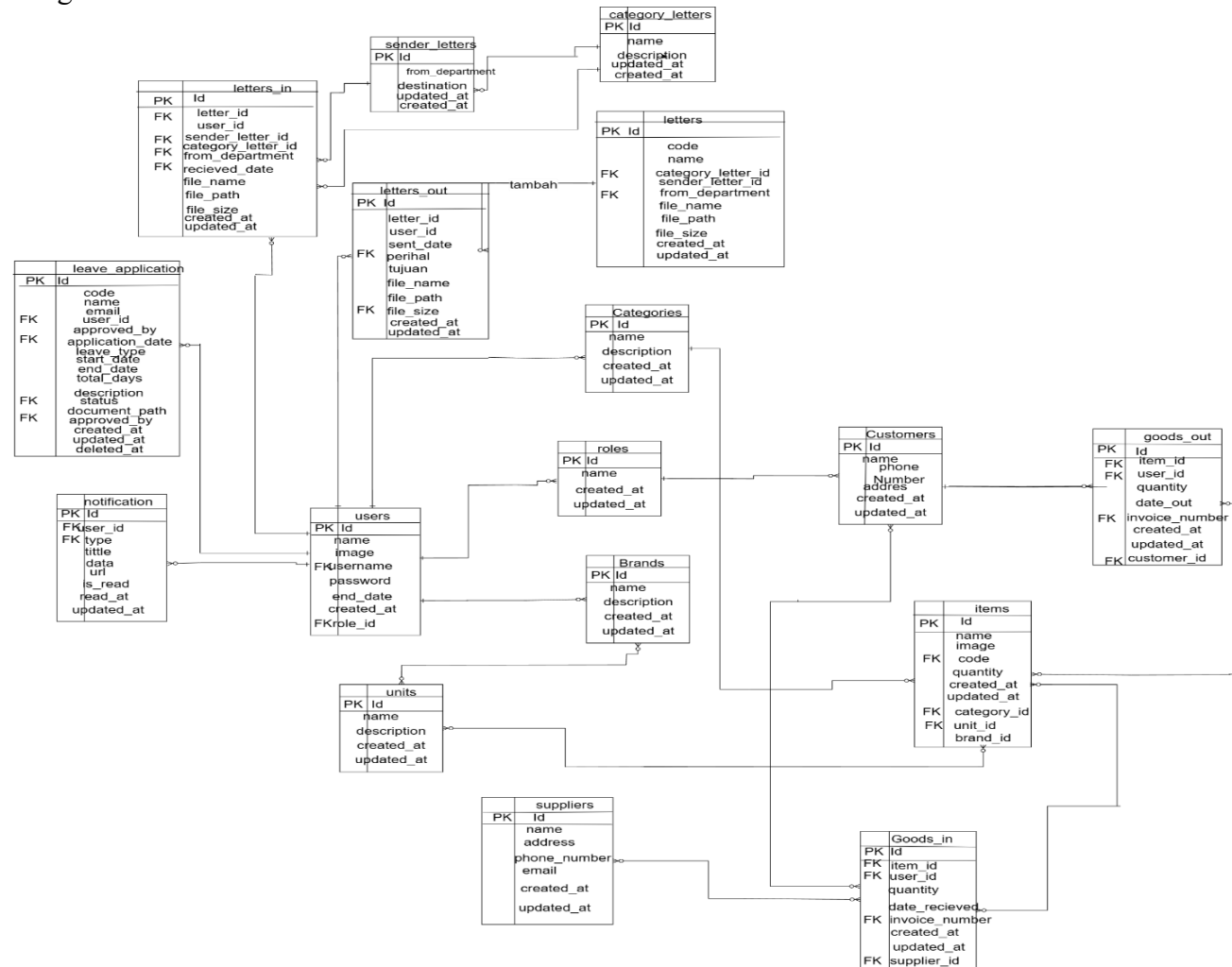
j. *Activity diagram* laporan stok barang



Gambar 3. 16 *Activity diagram* laporan stok barang

Berdasarkan gambar 3.21 di atas, *activity diagram* menggambarkan alur proses pembuatan laporan stok barang yang melibatkan tiga entitas yaitu Pengguna (Administrator), sistem, dan *database*. Proses dimulai ketika administrator mengakses halaman laporan stok barang, yang kemudian memicu pengambilan data barang masuk dari *database*. Data yang diambil kemudian digunakan sistem untuk menampilkan halaman laporan stok barang. Setelah halaman ditampilkan, terdapat *decision point* apakah melakukan cetak laporan atau tidak; jika Ya, Administrator klik tombol Cetak PDF/Excel yang dilanjutkan dengan sistem mencetak laporan dalam format PDF atau Excel, namun jika tidak melakukan cetak, sistem tetap menampilkan halaman laporan stok barang stok barang adalah untuk menampilkan data inventori terkini dan menyediakan opsi untuk mengekspor data dalam format yang dapat dicetak.

3.3.2.7 Perancangan database



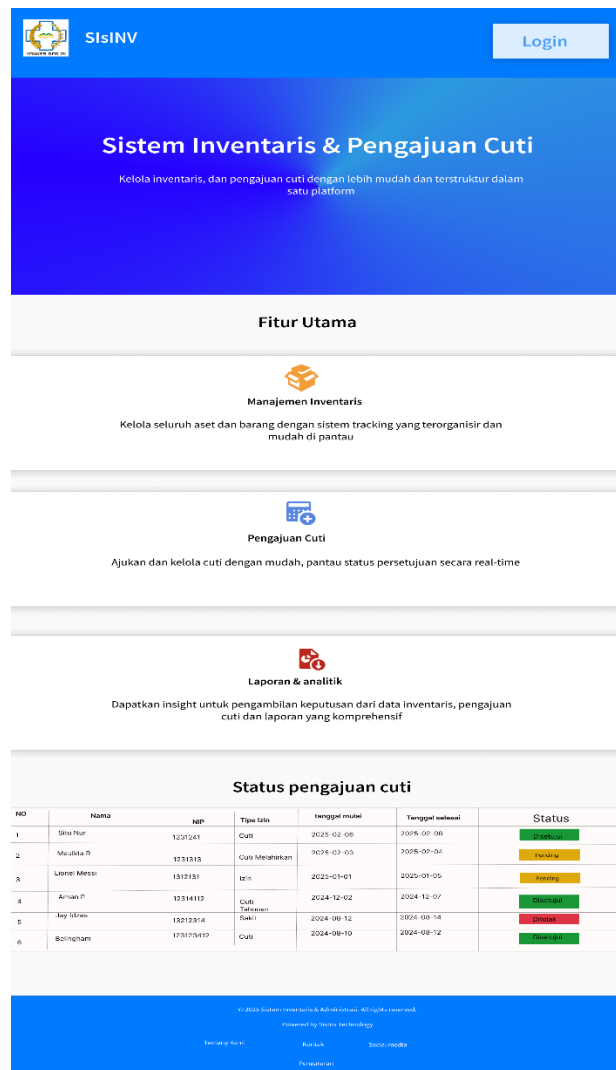
gambar 3. 17 Entity Relational Diagram Dashboard

Berdasarkan gambar ERD di atas terdiri dari 17 tabel yang saling berelasi dalam satu database yaitu: *leave_application*, *letters_in*, *letters_out*, *letters*, *categories*, *notification*, *users*, *roles*, *brands*, *customers*, *goods_out*, *items*, *units*, *suppliers*, dan *goods_in*. Setiap tabel memiliki *primary key* (PK) yang ditandai dengan "id" dan beberapa tabel memiliki *foreign key* (FK) yang menghubungkan dengan tabel lain. Sistem ini menggabungkan dua modul utama yaitu manajemen surat/dokumen dan sistem inventaris barang.

Dalam relasi *database* yang digunakan menunjukkan hubungan *one-to-many* yang dominan. Misalnya, tabel "*users*" berelasi *one-to-many* dengan "*letters_in*", "*letters_out*", "*goods_out*", dan "*notification*" melalui FK *user_id*, artinya satu user bisa memiliki banyak surat masuk/keluar dan transaksi barang. Tabel "*categories*" juga berelasi *one-to-many* dengan "*letters*" melalui FK *category_letter_id*. Untuk tabel "*items*", terdapat *multiple* FK yaitu *category_id*, *unit_id*, dan *brand_id* yang menunjukkan bahwa satu item memiliki satu kategori, satu unit, dan satu *brand* (relasi *many-to-one* dari *database* table *items*).

3.3.2.8 Design Interface

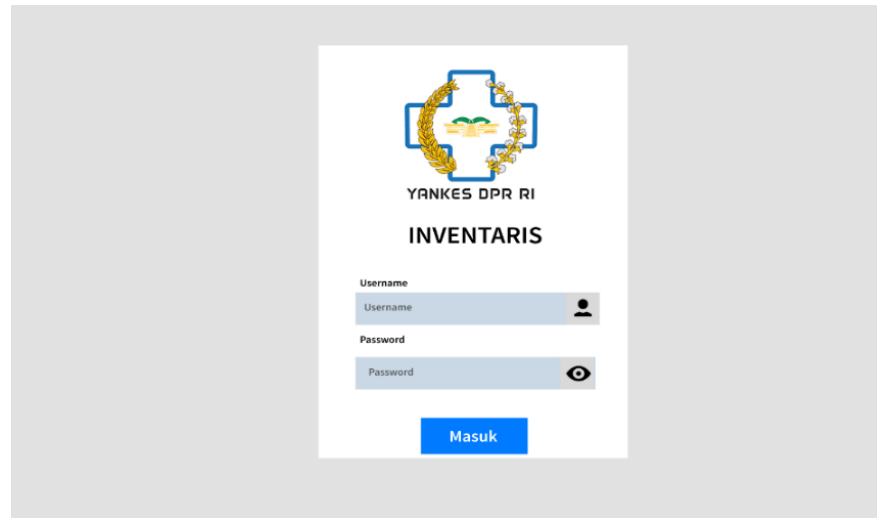
a. Desain landing page



Gambar 3. 18 Desain *mockup* halaman *landing page*

berdasarkan gambar di atas, tampilan adalah halaman utama dari sistem bernama "SisINV" yang berfungsi sebagai Sistem Inventaris dan Pengajuan Cuti. *Platform* ini menggabungkan dua fungsi utama yaitu manajemen inventaris dan sistem pengajuan cuti karyawan dalam satu aplikasi terintegrasi. dengan fitur utama yang jelas (manajemen inventaris, pengajuan cuti, dan laporan & analitik), serta tampilan status pengajuan cuti yang komprehensif (menampilkan nama, NIP, tipe izin, tanggal mulai, tanggal selesai, dan status persetujuan dengan warna yang membedakan status "Disetujui" , "Diproses" dan "Ditolak")

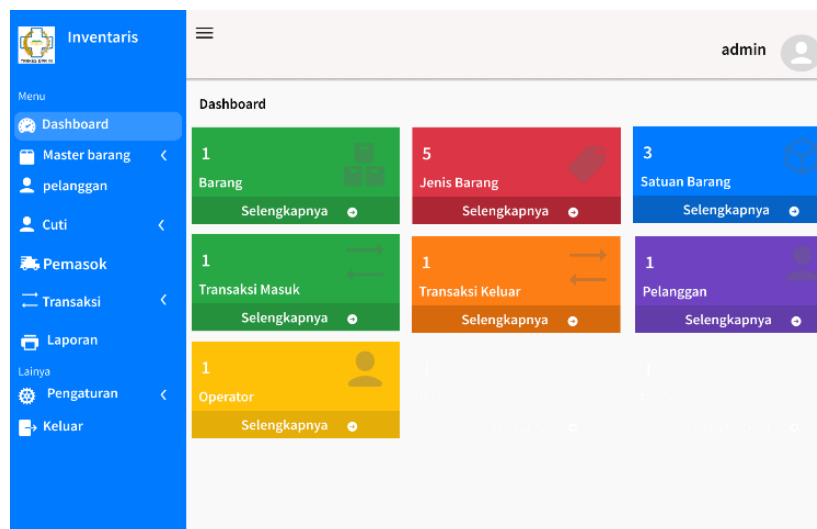
b. Desain *mockup* halaman *login*



Gambar 3. 19 Desain *mockup* halaman *login*

Berdasarkan gambar 3.24 halaman *login* untuk sistem inventaris Yankes DPR RI Halaman ini menampilkan logo institusi diikuti dengan tulisan "YANKES DPR RI" dan judul "INVENTARIS". *Form login* terdiri dari dua kolom input: kolom *username* dengan ikon pengguna dan kolom *password* dengan ikon mata untuk menampilkan/menyembunyikan *password*

c. Desain *mockup* halaman *dashboard*

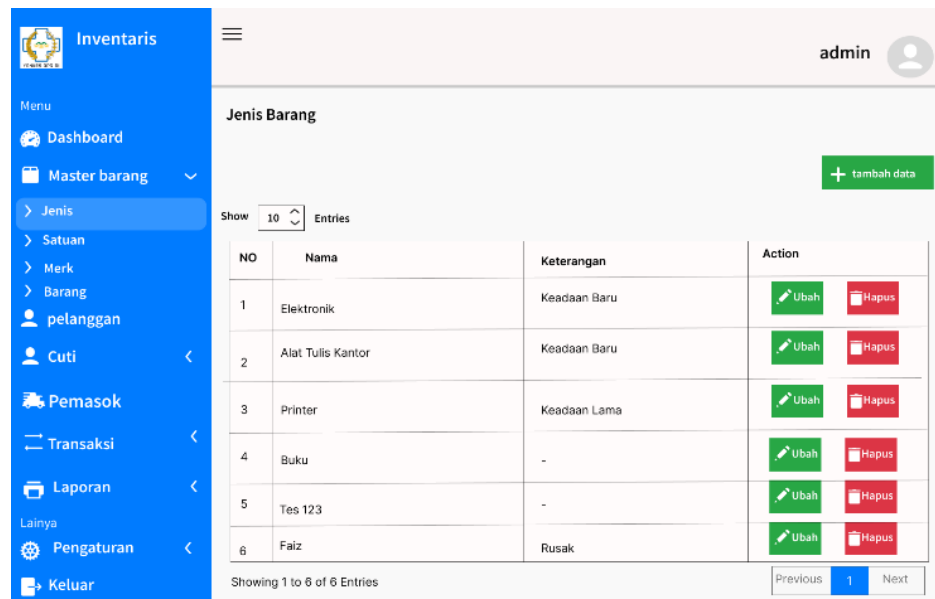


Gambar 3. 20 Halaman utama *dashboard*

berdasarkan gambar 3.25 di atas merupakan halaman utama Sistem sebelah kiri terdapat panel navigasi berwarna biru dengan logo aplikasi di bagian atas, diikuti dengan menu-menu seperti *dashboard*, Master Barang, Pelanggan, Cari, Pemasok,

Transaksi, Laporan, dan submenu Lainnya yang mencakup Pengaturan dan Keluar. Bagian utama dasbor menampilkan beberapa kartu informasi yang menunjukkan ringkasan data penting. Kartu-kartu tersebut meliputi: kartu hijau untuk Barang, kartu merah untuk Jenis Barang, kartu biru untuk Satuan Barang, kartu hijau lain untuk Transaksi Masuk, kartu oranye untuk Transaksi Keluar, kartu ungu untuk Pelanggan, dan kartu kuning untuk Operator.

d. Desain *mockup* halaman Jenis barang



The screenshot displays the 'Jenis Barang' (Goods Type) management page. It features a sidebar menu on the left with various navigation options. The main content area shows a table with 6 entries. Each entry includes a number (NO), a name (Nama), a description (Keterangan), and an action (Action) column with 'Ubah' (Edit) and 'Hapus' (Delete) buttons. A '+ tambah data' button is located at the top right of the table area. The table is paginated, showing 1 to 6 of 6 entries.

| NO | Nama | Keterangan | Action |
|----|-------------------|--------------|--|
| 1 | Elektronik | Keadaan Baru | Ubah Hapus |
| 2 | Alat Tulis Kantor | Keadaan Baru | Ubah Hapus |
| 3 | Printer | Keadaan Lama | Ubah Hapus |
| 4 | Buku | - | Ubah Hapus |
| 5 | Tes 123 | - | Ubah Hapus |
| 6 | Faiz | Rusak | Ubah Hapus |

Gambar 3. 21 Jenis barang

berdasarkan gambar 3.26 halaman "Jenis Barang" dari aplikasi web Inventaris. Bagian utama halaman menampilkan tabel data jenis barang dengan *header* berisi kolom-kolom: no, nama, keterangan, *action*. Tabel tersebut menampilkan 6 entri jenis (Keadaan Baru, Keadaan Lama, atau Rusak). Setiap baris dilengkapi tombol aksi *Edit* (hijau) dan *Hapus* (merah) di kolom *Action*

e. Desain mockup halaman satuan barang

Satuan Barang

+ tambah data

Show 10 Entries

| NO | Nama | Keterangan | Action |
|----|---------|------------|--|
| 1 | Rim | - | Ubah Hapus |
| 2 | License | lisensi | Ubah Hapus |
| 3 | Pcs | Pcs | Ubah Hapus |
| 4 | Unit | - | Ubah Hapus |
| 5 | Barang | - | Ubah Hapus |
| 6 | Test | - | Ubah Hapus |

Showing 1 to 6 of 6 Entries

Previous 1 Next

Gambar 3. 22 Satuan barang

berdasarkan gambar 3.27 halaman "Satuan Barang" dari aplikasi web Inventaris. Bagian utama halaman menampilkan tabel data satuan barang dengan kolom no, nama, Keterangan, dan *action*, yang berisi 6 entri satuan: Rim, *license*, pcs, unit, Barang, dan *test*, dengan tombol "Ubah" (hijau) dan "Hapus" (merah) untuk setiap baris.

f. Desain *mockup* halaman merek barang

Merk Barang

+ tambah data

Show 10 Entries

| NO | Nama | Keterangan | Action |
|----|----------|---------------|--|
| 1 | Asus | Asus X411U | Ubah Hapus |
| 2 | HP | HP Omen 152 | Ubah Hapus |
| 3 | Canon | Printer Canon | Ubah Hapus |
| 4 | Epson | Printer Epson | Ubah Hapus |
| 5 | Lenovo | Lenovo Legion | Ubah Hapus |
| 6 | Gramedia | 1 Rim Kertas | Ubah Hapus |

Showing 1 to 6 of 6 Entries

Previous 1 Next

Gambar 3. 23 Merk barang

berdasarkan gambar 3.28 Desain *mockup* halaman "Merek Barang" dari aplikasi web Inventaris. Bagian utama halaman menampilkan tabel data merek barang dengan kolom no, Nama, Keterangan, dan *action* yang berisi 6 entri setiap baris dilengkapi tombol "Ubah" (hijau) dan "Hapus" (merah)

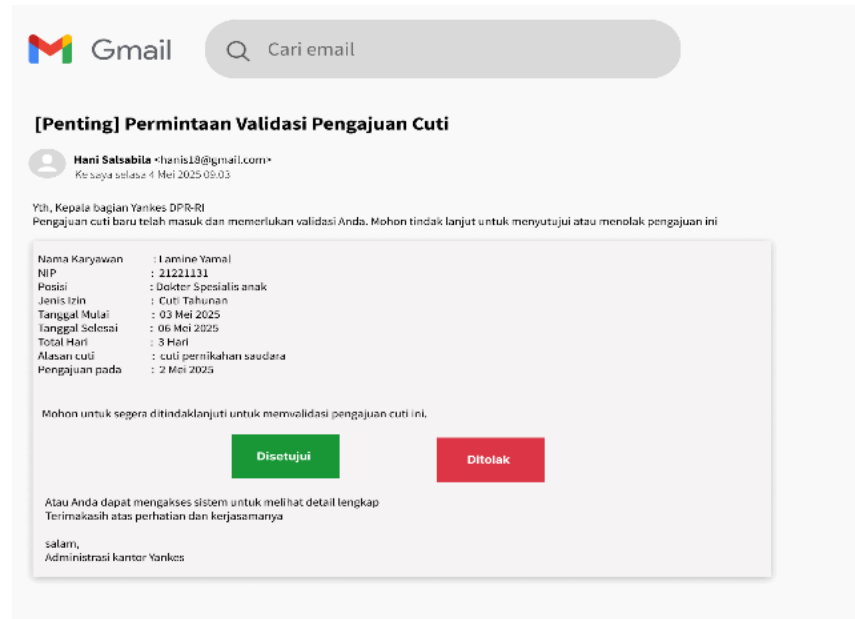
g. Desain mockup halaman barang

| NO | Foto | Kode | Nama | Jenis | Satuan | Merk | Stock Awal | Action |
|----|------|-------------|---------------|-------------------|--------|----------|------------|--|
| 1 | | BRG-1329191 | Printer | Elektronik | unit | Epson | 2 | Ubah Hapus |
| 2 | | BRG-1329192 | Tempat pensil | Alat Tulis kantor | pcs | Gramedia | 7 | Ubah Hapus |
| 3 | | BRG-1329194 | Kamera | Elektronik | unit | Canon | 1 | Ubah Hapus |
| 4 | | BRG-1329195 | Hvs | Alat Tulis kantor | Rim | Gramedia | 10 | Ubah Hapus |
| 5 | | BRG-1329196 | Laptop | Elektronik | unit | Asus | 5 | Ubah Hapus |
| 6 | | BRG-1329197 | mouse | Elektronik | unit | Asus | 6 | Ubah Hapus |

Gambar 3. 24 Barang

berdasarkan gambar 3.29 halaman barang *dashboard* sistem inventaris. Pada bagian halaman terdapat tabel data barang yang mencakup kolom seperti no, foto, Kode, Nama, Jenis, Satuan, Merek, dan stok Awal, serta kolom *action* yang menyediakan tombol hijau "Ubah" dan tombol merah "Hapus"

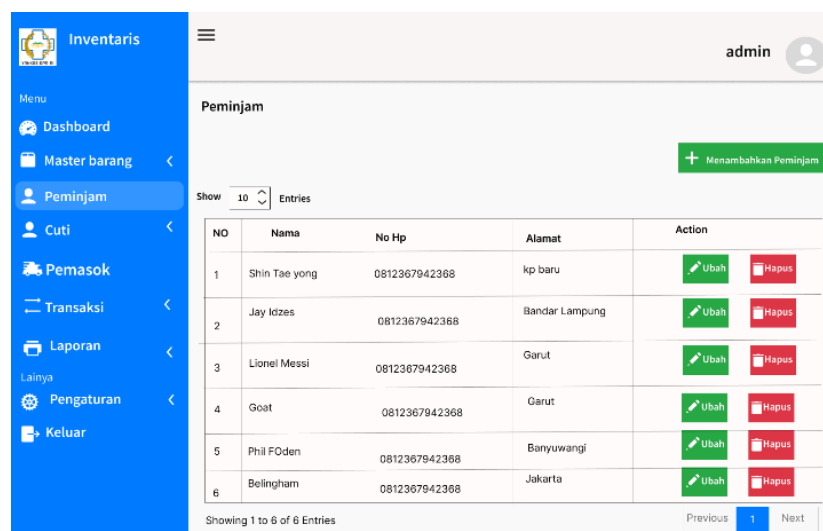
h. Desain *mockup* halaman notifikasi email



Gambar 3. 25 Desain *mockup* halaman notifikasi email

pada gambar 3.30 di atas halaman notifikasi email halaman tersebut terdapat pengiriman email kepada kepala bagian untuk memvalidasi pengajuan cuti yang telah diajukan. Pada proses pengiriman notifikasi email menggunakan protokol SMTP meliputi tahapan seperti 1) konfigurasi SMTP *server* dengan parameter autentikasi dan enkripsi 2) perancangan *template email* dengan format HTML/teks 3) gun *library* pengiriman email seperti *PHPMailer* untuk *library* pengiriman email dalam *PHP*

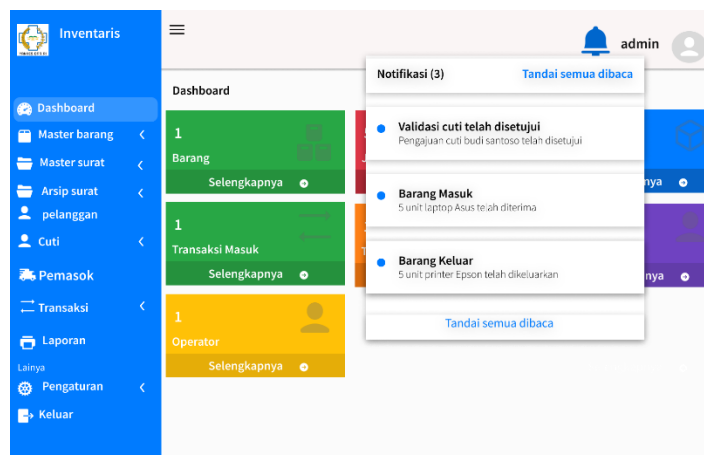
i. Desain *mockup* halaman peminjam



Gambar 3. 26 Halaman peminjam

Berdasarkan gambar 3.31, halaman Peminjam pada sistem inventaris ini menampilkan daftar peminjam barang dalam bentuk tabel yang terdiri dari kolom no, Nama, no HP, Alamat, serta *Action*. Tabel ini memungkinkan pengguna untuk melihat informasi peminjam secara jelas dan terstruktur. Di bagian atas, terdapat tombol hijau bertuliskan "+ Menambahkan Peminjam" yang memungkinkan pengguna untuk menambahkan data peminjam baru. Pada kolom *Action*, tersedia tombol hijau "Ubah" untuk mengedit data peminjam dan tombol merah "Hapus" untuk menghapus data peminjam

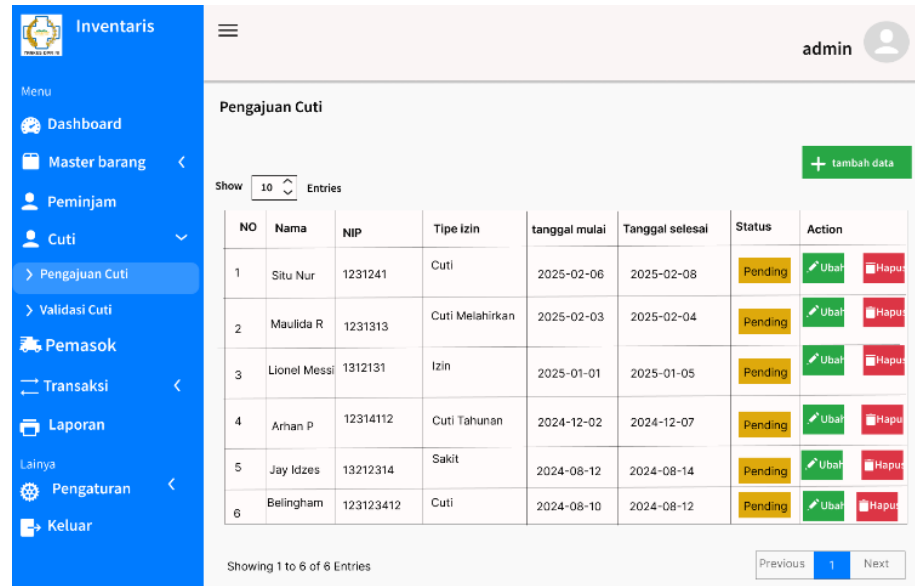
j. Desain mockup halaman notifikasi



Gambar 3. 27 Desain *mockup* halaman notifikasi

berdasarkan gambar 3.32 di atas merupakan gambar dari halaman notifikasi dalam sistem informasi inventaris yang berfungsi untuk memberi tahu admin mengenai aktivitas penting secara *real-time*. Notifikasi muncul di bagian kanan atas melalui ikon lonceng dan menampilkan daftar informasi seperti validasi cuti yang telah disetujui, barang masuk, dan barang keluar, lengkap dengan deskripsi singkat untuk masing-masing peristiwa. Notifikasi yang belum dibaca ditandai dengan titik biru, sementara di bagian atas terdapat opsi "Tandai semua dibaca"

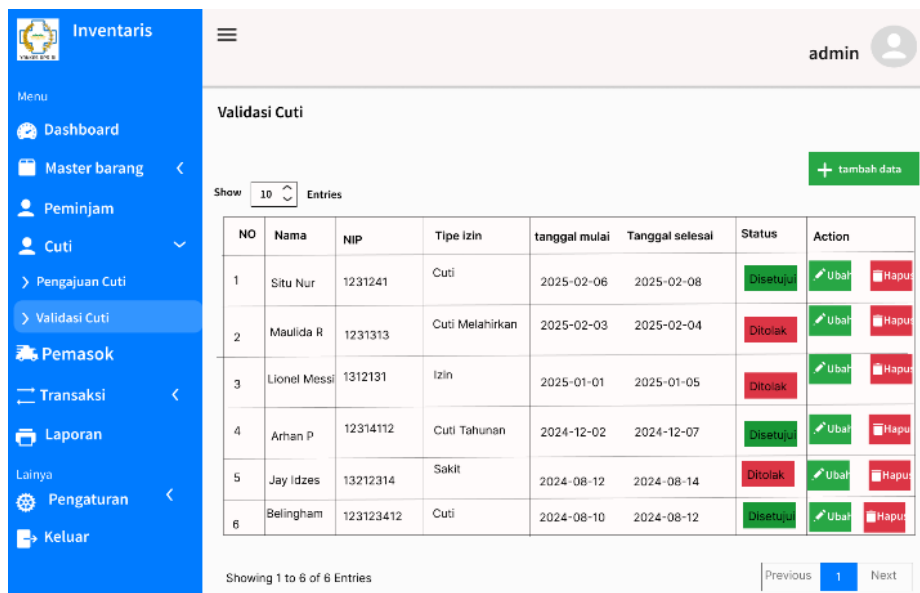
k. Desain *mockup* halaman pengajuan cuti



Gambar 3. 28 Halaman pengajuan cuti

Berdasarkan gambar 3.33 halaman Pengajuan Cuti pada sistem inventaris ini menampilkan daftar pegawai yang mengajukan cuti dalam bentuk tabel yang terdiri dari kolom no, nama, NIP, Tipe Izin, Tanggal Mulai, Tanggal Selesai, Status, dan Action. Setiap entri dalam tabel memiliki status "Pending" yang ditampilkan dalam label berwarna kuning, menandakan bahwa pengajuan cuti masih menunggu persetujuan.

1. Desain *mockup* halaman validasi cuti



Gambar 3. 29 Halaman validasi cuti

Berdasarkan gambar 3.34 halaman Pengajuan Cuti pada sistem inventaris ini menampilkan daftar pengajuan cuti pegawai dalam bentuk tabel yang terdiri dari kolom No, Nama, NIP, Tipe Izin, Tanggal Mulai, Tanggal Selesai, Status, dan Action. Pada tabel ini, status pengajuan cuti dapat berupa "Disetujui" yang ditampilkan dalam label hijau atau "Ditolak" yang ditampilkan dalam label merah, menunjukkan hasil validasi cuti yang telah diproses. Di bagian atas halaman, terdapat tombol hijau "+ Tambah Data" untuk menambahkan pengajuan cuti baru. Sementara itu, pada kolom *Action*, tersedia tombol hijau "Ubah" untuk mengedit data pengajuan cuti dan tombol merah "Hapus" untuk menghapus data tersebut

m. Desain mockup halaman pemasok

| NO | Nama | No Hp | Alamat | Email | Action |
|----|---------------|------------|----------------|---------------------------|--|
| 1 | Shin Tae yong | 0821821021 | kp baru | Shintaeyoung123@gmail.com | Ubah Hapus |
| 2 | Jay Idzes | 08128191 | Bandar Lampung | Jayldzen44@gmail.com | Ubah Hapus |
| 3 | Lionel Messi | 08219102 | Garut | MessiGoat01@gmail.com | Ubah Hapus |
| 4 | Goat | 08192910 | Garut | GoatMessi10@gmail.com | Ubah Hapus |
| 5 | Phil Foden | 08129190 | Banyuwangi | Foden33@gmail.com | Ubah Hapus |
| 6 | Bellingham | 08131920 | Jakarta | Belingoal@gmail.com | Ubah Hapus |

Gambar 3. 30 Halaman pemasok

Berdasarkan gambar 3.35 halaman Pemasok, tampilan aplikasi web Inventaris halaman utama Pemasok yang sedang aktif ditandai dengan latar belakang berbeda. Bagian utama halaman menampilkan tabel data pemasok dengan struktur kolom yang lebih kompleks, terdiri dari no, nama, no HP, Alamat, *Email*, dan *action*. Tabel berisi

n. Desain *mockup* halaman transaksi masuk

| NO | tanggal masuk | kode barang masuk | kode barang | Pemasok | Jumlah masuk | Action |
|----|---------------|-------------------|-------------|--------------|--------------|--|
| 1 | 2025-02-06 | BRGMSK-12231139 | BRG-1329191 | Shin Tae yon | 3/120 | Ubah Hapus |
| 2 | 2025-02-03 | BRGMSK-414124 | BRG-1329192 | Jay Idzes | 3/16 | Ubah Hapus |
| 3 | 2025-01-01 | BRGMSK-1241255 | BRG-1329194 | Lionel Messi | 3/17 | Ubah Hapus |
| 4 | 2024-12-02 | BRGMSK-23242 | BRG-1329195 | Goat | 3/10 | Ubah Hapus |
| 5 | 2024-08-12 | BRGMSK-122311233 | BRG-1329196 | Phil Foden | 3/12 | Ubah Hapus |
| 6 | 2024-08-10 | BRGMSK-14342532 | BRG-1329197 | Bellingham | 3/11 | Ubah Hapus |

Gambar 3. 31 Halaman transaksi masuk

Berdasarkan gambar 3.36 halaman transaksi masuk, tampilan aplikasi web Inventaris menunjukkan Bagian utama halaman menampilkan tabel data transaksi masuk dengan struktur kolom yang terdiri dari no, tanggal masuk, kode barang masuk, kode barang, Pemasok, Jumlah masuk, dan *action*. serta tombol aksi "Ubah" dan "Hapus".

o. Desain *mockup* halaman transaksi keluar

| NO | tanggal Keluar | kode barang | kode barang | Peminjam | Jumlah Keluar | Action |
|----|----------------|------------------|-------------|--------------|---------------|--|
| 1 | 2025-02-06 | BRGKLR-12231139 | BRG-1329191 | Shin Tae yon | 3/120 | Ubah Hapus |
| 2 | 2025-02-03 | BRGKLR-414124 | BRG-1329192 | Jay Idzes | 3/16 | Ubah Hapus |
| 3 | 2025-01-01 | BRGKLR-1241255 | BRG-1329194 | Lionel Messi | 3/17 | Ubah Hapus |
| 4 | 2024-12-02 | BRGKLR-23242 | BRG-1329195 | Goat | 3/10 | Ubah Hapus |
| 5 | 2024-08-12 | BRGKLR-122311233 | BRG-1329196 | Phil Foden | 3/12 | Ubah Hapus |
| 6 | 2024-08-10 | BRGKLR-14342532 | BRG-1329197 | Bellingham | 3/11 | Ubah Hapus |

Gambar 3. 32 Halaman transaksi masuk

Berdasarkan gambar 3.37 halaman Transaksi Keluar, bagian utama halaman Transaksi Keluar yang aktif di bawah menu utama "Transaksi". Bagian utama halaman menampilkan tabel data transaksi keluar dengan struktur kolom yang mirip

dengan halaman transaksi masuk, terdiri dari no, tanggal keluar, kode barang, kode barang (kolom duplikat), Peminjam, Jumlah Keluar, dan *action*.

p. Desain *mockup* halaman laporan barang masuk

| NO | tanggal | kode barang | kode barang | Pemasok | Barang | Jumlah Masuk |
|----|------------|------------------|-------------|--------------|-------------------|--------------|
| 1 | 2025-02-06 | BRGMSK-12231139 | BRG-1329191 | Shin Tae yon | Elektronik | 3/120 |
| 2 | 2025-02-03 | BRGMSK-414124 | BRG-1329192 | Jay Idzes | Alat Tulis kantor | 3/16 |
| 3 | 2025-01-01 | BRGMSK-1241255 | BRG-1329194 | Lionel Messi | Elektronik | 3/17 |
| 4 | 2024-12-02 | BRGMSK-23242 | BRG-1329195 | Goat | Alat Tulis kantor | 3/10 |
| 5 | 2024-08-12 | BRGMSK-122311233 | BRG-1329196 | Phil Foden | Elektronik | 3/12 |
| 6 | 2024-08-10 | BRGMSK-14342532 | BRG-1329197 | Bellingham | Elektronik | 3/11 |

Gambar 3. 33 Halaman barang masuk

Berdasarkan gambar 3.38 halaman laporan barang masuk, Bagian utama halaman menampilkan tabel data laporan barang masuk dengan struktur kolom yang lebih informatif, terdiri dari no, tanggal, kode barang, kode barang (kolom duplikat), pemasok, barang, dan jumlah masuk. Tabel berisi 6 entri dengan informasi tanggal yang sama seperti di halaman transaksi tombol ekspor data di atas tabel, meliputi "Cetak *Print*", "Cetak *PDF*", dan "Cetak *Excel*"

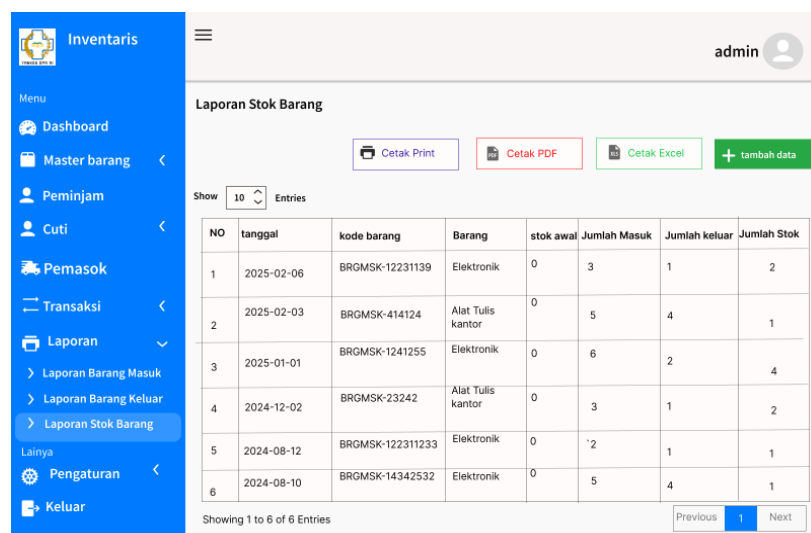
q. Desain *mockup* halaman laporan barang keluar

| NO | tanggal | kode barang | kode barang | Peminjam | Barang | Jumlah |
|----|------------|------------------|-------------|--------------|-------------------|--------|
| 1 | 2025-02-06 | BRGMSK-12231139 | BRG-1329191 | Shin Tae yon | Elektronik | 3/120 |
| 2 | 2025-02-03 | BRGMSK-414124 | BRG-1329192 | Jay Idzes | Alat Tulis kantor | 3/16 |
| 3 | 2025-01-01 | BRGMSK-1241255 | BRG-1329194 | Lionel Messi | Elektronik | 3/17 |
| 4 | 2024-12-02 | BRGMSK-23242 | BRG-1329195 | Goat | Alat Tulis kantor | 3/10 |
| 5 | 2024-08-12 | BRGMSK-122311233 | BRG-1329196 | Phil Foden | Elektronik | 3/12 |
| 6 | 2024-08-10 | BRGMSK-14342532 | BRG-1329197 | Bellingham | Elektronik | 3/11 |

Gambar 3. 34 Halaman laporan barang keluar

Berdasarkan gambar 3.39, halaman "Laporan Barang Keluar" menampilkan desain antarmuka sistem inventaris. Pada sisi kiri, terdapat menu navigasi berwarna biru. Bagian utama halaman menampilkan tabel laporan barang keluar yang terdiri dari kolom nomor, tanggal, kode barang, peminjam, jenis barang, dan jumlah barang. Di atas tabel, terdapat tombol aksi untuk mencetak laporan dalam berbagai format, seperti *Print*, PDF, dan Excel, serta tombol hijau "+ Tambah Data" untuk menambahkan entri baru.

r. Desain *mockup* halaman laporan stok barang

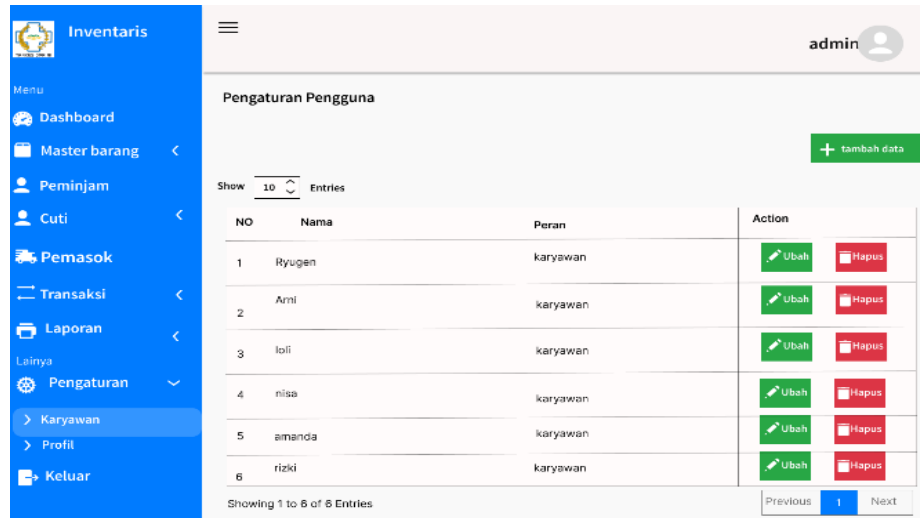


| NO | tanggal | kode barang | Barang | stok awal | Jumlah Masuk | Jumlah keluar | Jumlah Stok |
|----|------------|------------------|-------------------|-----------|--------------|---------------|-------------|
| 1 | 2025-02-06 | BRGMSK-12231139 | Elektronik | 0 | 3 | 1 | 2 |
| 2 | 2025-02-03 | BRGMSK-414124 | Alat Tulis kantor | 0 | 5 | 4 | 1 |
| 3 | 2025-01-01 | BRGMSK-1241255 | Elektronik | 0 | 6 | 2 | 4 |
| 4 | 2024-12-02 | BRGMSK-23242 | Alat Tulis kantor | 0 | 3 | 1 | 2 |
| 5 | 2024-08-12 | BRGMSK-122311233 | Elektronik | 0 | 2 | 1 | 1 |
| 6 | 2024-08-10 | BRGMSK-14342532 | Elektronik | 0 | 5 | 4 | 1 |

Gambar 3. 35 Halaman laporan stok barang

Berdasarkan gambar 3.40, halaman Laporan Stok Barang dalam sistem inventaris untuk mengelola data stok barang. Pada bagian utama, terdapat tabel yang menyajikan informasi mengenai stok barang, termasuk nomor, tanggal, kode barang, nama barang, stok awal, jumlah masuk, jumlah keluar, dan jumlah stok tersisa. Di atas tabel, tersedia tombol aksi untuk mencetak laporan dalam format *Print*, PDF, dan Excel, serta tombol hijau "+ Tambah Data".

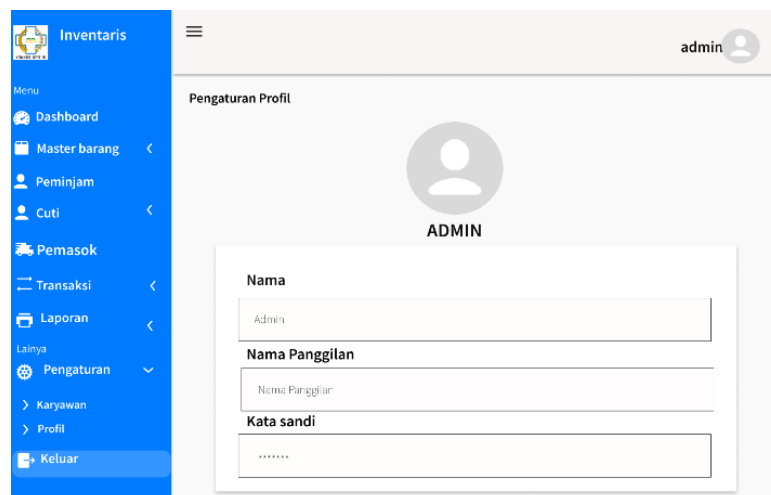
s. Desain *mockup* halaman pengaturan pengguna



Gambar 3. 36 Halaman pengaturan pengguna

Berdasarkan gambar 3.41, halaman Pengaturan Pengguna pada sistem inventaris ini dirancang untuk mengelola data pengguna. Pada bagian utama, terdapat tabel yang menampilkan daftar pengguna dengan kolom nomor, nama, peran, dan aksi. Setiap baris memiliki dua tombol aksi, yaitu tombol hijau "Ubah" untuk mengedit data pengguna dan tombol merah "Hapus" untuk menghapus data

t. Desain *mockup* halaman profil



Gambar 3. 37 Halaman Profile

Berdasarkan gambar 3.42, halaman "pengaturan profil" dalam sistem inventaris ini dirancang untuk memungkinkan pengguna mengelola informasi pribadi mereka. Pada bagian utama halaman, terdapat tampilan profil pengguna dengan ikon avatar besar dan label "ADMIN," yang menunjukkan bahwa pengguna saat ini memiliki

peran sebagai administrator. Di bawahnya, terdapat formulir pengaturan profil yang terdiri dari bidang input untuk Nama, Nama Panggilan, dan Kata Sandi.

3.3.2.9 Design Testing (System Usability Scale)

metode SUS digunakan untuk mengukur tingkat kepuasan pengguna terhadap desain sistem yang telah dikembangkan. Pengujian dilakukan dengan melibatkan sejumlah responden yang diberikan akses desain sistem untuk melakukan berbagai tugas yang telah ditentukan. Setelah itu, mereka diminta untuk mengisi kuesioner SUS, yang terdiri dari sepuluh pernyataan yang mencerminkan aspek kemudahan penggunaan, efisiensi, dan kenyamanan dalam berinteraksi dengan sistem.

Tabel 3. 7 Hasil Perhitungan *system usability scale*

| No | Responden | Posisi jabatan | Skala Hasil Hitung | | | | | | | | | | Jumlah | Nilai (Jumlah x 2.5) |
|-------------------------------|--------------------|----------------------|--------------------|-----|-----|-----|-----|-----|-----|-----|-----|------|--------|-------------------------|
| | | | Q#1 | Q#2 | Q#3 | Q#4 | Q#5 | Q#6 | Q#7 | Q#8 | Q#9 | Q#10 | | |
| 1 | Hendra | Gudang dan Ambulance | 3 | 3 | 4 | 2 | 3 | 3 | 3 | 3 | 4 | 1 | 29 | 72.5 |
| 2 | Paini | Gudang | 4 | 3 | 4 | 4 | 4 | 3 | 4 | 4 | 4 | 4 | 37 | 92.5 |
| 3 | Ghassani Sophie | gudang obat | 3 | 4 | 3 | 4 | 4 | 4 | 3 | 3 | 3 | 2 | 33 | 82.5 |
| 4 | Wati | Administrasi | 3 | 4 | 4 | 3 | 3 | 4 | 3 | 3 | 3 | 3 | 33 | 82.5 |
| 5 | Widya | Administrasi | 3 | 3 | 3 | 3 | 3 | 3 | 4 | 3 | 3 | 3 | 31 | 77.5 |
| 6 | Ichsan Nugroho | Administrasi | 3 | 4 | 3 | 4 | 4 | 3 | 3 | 4 | 3 | 4 | 35 | 87.5 |
| 7 | dr. Dian Mardiyani | Kepala bagian Yankes | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 40 | 100 |
| 8 | Rio Ariesta | Dosen ahli IT | 3 | 3 | 0 | 4 | 3 | 3 | 2 | 3 | 3 | 3 | 27 | 67.5 |
| 9 | Resty Annisa | Dosen ahli IT | 3 | 3 | 3 | 0 | 3 | 0 | 3 | 3 | 3 | 3 | 24 | 60 |
| 10 | Trisya Septiana | Dosen ahli IT | 3 | 3 | 2 | 4 | 0 | 3 | 0 | 4 | 0 | 3 | 22 | 55 |
| Skala Rata-rata (Hasil Akhir) | | | | | | | | | | | | | 77.75 | |

$$\text{Total Skor} = \Sigma(\text{Skor Item 1} + \text{Skor Item 2} + \dots + \text{Skor Item 10}) \quad (1)$$

$$\text{SUS Score} = \text{Total Skor} \times 2.5$$

Berdasarkan hasil tabel 3.7 dari pengujian SUS di atas perhitungan skala SUS dilakukan dengan mengonversi jawaban menjadi skala numerik, proses perhitungan untuk pernyataan bernomor ganjil (1,3,5,7,9), kurangi 1 dari jawaban pengguna; untuk pernyataan bernomor genap (2,4,6,8,10), kurangi jawaban pengguna dari 5. menjumlahkan semua skor yang telah dihitung untuk setiap responden. Selanjutnya mengalikan total skor dengan 2,5 untuk mendapatkan nilai *SUS* akhir pada skala 0-100. Terakhir, menginterpretasi hasil dengan panduan, skor di atas 68 dianggap di atas rata-rata (baik), skor 85 ke atas dianggap sangat baik, hasil *questioner SUS* di atas mendapatkan nilai rata-rata yaitu 77.75. Berdasarkan hasil skala penilaian *system usability scale* dapat dikategorikan sebagai sistem ini dalam kategori *Excellent* atau lebih dari baik.

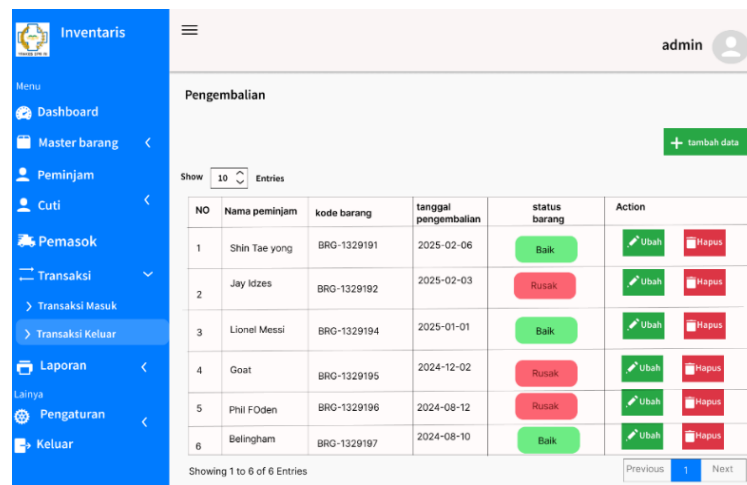
Tabel 3. 8 Skala Interpretasi Hasil Skor SUS [29]

| Grade | SUS | Percentile range | Adjective | Acceptable | NPS |
|--------------|-------------|-----------------------------|------------------------|-------------------|------------------|
| A+ | 84.1 – 100 | 96 – 100 | <i>Best Imaginable</i> | <i>Acceptable</i> | <i>Promoter</i> |
| A | 80.8 – 84.0 | 90 – 95 | <i>Excellent</i> | <i>Acceptable</i> | <i>Promoter</i> |
| A- | 78.9 – 80.7 | 85 – 89 | <i>Good</i> | <i>Acceptable</i> | <i>Promoter</i> |
| B+ | 77.2 – 78.8 | 80 – 84 | | <i>Acceptable</i> | <i>Passive</i> |
| B | 74.1 – 77.1 | 70 – 79 | | <i>Acceptable</i> | <i>Passive</i> |
| B- | 72.6 – 74.0 | 65 – 69 | | <i>Acceptable</i> | <i>Passive</i> |
| C+ | 71.1 – 72.5 | 60 – 64 | | <i>Acceptable</i> | <i>Passive</i> |
| C | 65.0 – 71.0 | 41 – 59 | <i>OK</i> | <i>Marginal</i> | <i>Passive</i> |
| C- | 62.7 – 64.9 | 35 – 40 | | <i>Marginal</i> | <i>Passive</i> |
| D | 51.7 – 62.6 | 15 – 34 | | <i>Marginal</i> | <i>Detractor</i> |

1) Iterasi 1 *feedback system usability scale*

Terdapat masukan dari staf untuk menambahkan pengembalian barang dengan kondisi status barang pengembalian yang sudah selesai dipinjam oleh peminjam, berdasarkan Hasil masukan *SUS* di atas proses ini mengulang (iterasi) ke dalam tahapan *user design*, untuk melakukan perbaikan berdasarkan saran dari pengguna akhir sistem.

a. Desain *mockup* halaman pengembalian barang



| NO | Nama peminjam | kode barang | tanggal pengembalian | status barang | Action |
|----|---------------|-------------|----------------------|---------------|--|
| 1 | Shin Tae yong | BRG-1329191 | 2025-02-06 | Baik | Ubah Hapus |
| 2 | Jay Idzes | BRG-1329192 | 2025-02-03 | Rusak | Ubah Hapus |
| 3 | Lionel Messi | BRG-1329194 | 2025-01-01 | Baik | Ubah Hapus |
| 4 | Goat | BRG-1329195 | 2024-12-02 | Rusak | Ubah Hapus |
| 5 | Phil Foden | BRG-1329196 | 2024-08-12 | Rusak | Ubah Hapus |
| 6 | Bellingham | BRG-1329197 | 2024-08-10 | Baik | Ubah Hapus |

gambar 3. 38 Desain mockup halaman pengembalian barang

pada gambar 3.43 di atas halaman pengembalian barang dalam sistem inventaris, dimana admin dapat melihat dan mengelola data barang yang telah dikembalikan oleh para peminjam. Setiap entri menampilkan informasi penting seperti nama peminjam, kode barang, tanggal pengembalian, serta status barang apakah dalam kondisi "Baik" atau "Rusak" (ditandai dengan warna hijau dan merah untuk membedakan secara visual). Di sisi kanan, terdapat tombol "Ubah" untuk mengedit data pengembalian dan "Hapus"

Setelah implementasi perbaikan, dilakukan validasi ulang terhadap antarmuka dan alur proses pengembalian, untuk memastikan bahwa perubahan yang diterapkan telah sesuai dengan harapan pengguna. Pengujian ulang dilakukan secara internal, melibatkan pengguna akhir sistem untuk mengamati kembali aspek kemudahan penggunaan, efisiensi, serta kejelasan informasi yang ditampilkan.

3.4.1 Construction(implementasi)

Pada tahap implementasi ini, dilakukan berdasarkan Hasil analisis dan perancangan yang telah dilakukan pada tahap sebelumnya. Pengembangan dimulai dengan pembuatan kode program menggunakan *framework Laravel* dan *database MySQL*. Proses ini melibatkan pembangunan berbagai fitur fungsional, seperti manajemen barang, pengelolaan data administrasi, serta sistem pelaporan yang ada dalam *dashboard*, dalam perencanaan *hosting* sistem informasi di-*deploy* pada layanan *shared hosting* Niagahoster dengan kapasitas 25 GB *SSD storage* dan batasan hingga 400.000 *inode* (file dan direktori). Proses *deployment* dimulai dengan persiapan akun *hosting* dan pengaturan domain/subdomain melalui *CPanel*.

3.4.2 Cutover (pengujian)

Tahap *cutover* merupakan tahap akhir sebelum sistem diterapkan sepenuhnya dalam operasional Bagian Layanan Kesehatan DPR-RI. Pada tahap ini, sistem yang telah dibangun diuji secara menyeluruh dengan menggunakan dua jenis pengujian utama, yaitu *Black box testing* dan *User Acceptance Testing (UAT)*.

1. Black box testing

dilakukan untuk menguji fungsionalitas sistem tanpa memperhatikan struktur internal kode program. Fokus utama pengujian ini adalah untuk memastikan bahwa semua fitur yang ada, seperti pengelolaan barang, pencatatan transaksi, dan laporan administrasi, dapat berjalan dengan baik sesuai dengan kebutuhan fungsional yang telah ditetapkan sebelumnya. Pengujian ini melibatkan pengguna untuk memastikan bahwa sistem dapat menerima input dari pengguna dengan benar dan menghasilkan *output* yang sesuai tanpa adanya kesalahan.

2. User Acceptance Testing (UAT)

bertujuan untuk memastikan bahwa sistem yang dikembangkan sudah sesuai dengan harapan dan kebutuhan pengguna akhir. Pengujian ini melibatkan pihak pengguna, seperti staf Bagian Layanan Kesehatan DPR-RI, yang memvalidasi apakah sistem dapat digunakan dengan mudah dan

dapat mendukung tugas-tugas administrasi serta pengelolaan inventaris secara baik.

Setelah proses *cutover* ini, perbaikan dilakukan jika ditemukan masalah selama pengujian, dan sistem akhirnya disiapkan untuk digunakan oleh bagian Layanan Kesehatan DPR-RI dengan tujuan meningkatkan efisiensi dan akurasi dalam pengelolaan data administrasi dan inventaris barang.

V. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Adapun kesimpulan yang diperoleh dari penelitian yang telah dilakukan adalah sebagai berikut.

1. Penerapan metode *Rapid Application Development* dalam pembuatan *website* Portal Registrasi Inventaris dan Manajemen Administrasi Yankes DPR-RI menghasilkan 2 *role* dalam sistem yaitu *admin* dan *employee*.
2. Berdasarkan hasil pengujian *system usability scale* (SUS) menghasilkan 80% pada tahapan desain, sedangkan pada pengujian *black box testing* untuk menguji fungsionalitas sistem, proses *blackbox testing* menunjukkan adanya perbaikan sebanyak 2x (iterasi). Pada iterasi pertama menghasilkan nilai 94,37% pada prosesnya, setelah melakukan perbaikan pengujian menghasilkan nilai 100%, sedangkan *User Acceptance Testing* (UAT) menghasilkan nilai rata-rata 86% yang dikategorikan "Baik" dengan rincian: *Usability* 90% (Sangat Baik), *Functionality* 80% (Baik), dan *Efficiency* 84% (Baik) proses pembuatan sistem dapat membantu untuk inventaris dan data administrasi.
3. *Website* primasiyankes.id di *hosting* menggunakan *shared hosting* niaga hoster dengan penyimpanan sebesar 25 GB, sistem ini diperkirakan membutuhkan penyimpanan sebesar 3,6 GB per tahun untuk tiga jenis input data *file* surat (satuan surat, surat masuk, dan surat keluar), masing-masing input *file* sebesar 2MB, estimasi penggunaan data menunjukkan bahwa sistem mencapai kapasitas penuh (25 GB) pada tahun ke-6,5, dengan tingkat penggunaan penyimpanan mencapai 100%.

5.2.Saran

Berdasarkan hasil dari penelitian yang telah dilakukan, terdapat beberapa saran untuk pengembangan sistem selanjutnya, antara lain:

1. Menambahkan fitur *log* aktivitas pengguna untuk memantau setiap tindakan yang dilakukan oleh pengguna dan menambahkan *barcode scan* pada sistem untuk mempermudah proses pengelolaan inventaris dan transaksi barang serta menambahkan *button* validasi pengajuan cuti pada email kepala bagian.
2. Menambahkan kapasitas penyimpanan sistem, untuk membantu memaksimalkan jumlah data yang masuk ke dalam sistem, menambahkan kapasitas penyimpanan lebih atau menggunakan *server* jaringan internal yang sudah ada dari Yankes DPR-RI,
3. Menambahkan fitur berupa lihat dokumen dan unduh dokumen pada surat, pengajuan cuti, validasi cuti. Serta menambahkan fitur integrasi dengan sistem absensi Yankes DPR-RI.

DAFTAR PUSTAKA

- [1] N. L. Meilani, “Mal – Administrasi Dalam Penyelenggaraan Pengelolaan Perparkiran Di Kota Bukittinggi,” *Journal Unhas* Vol. 5, No. Edisi Ii Juli-Desember, Pp. 1–15, 2019.
- [2] R. Wahyudi, “Kelalaian Birokrasi Dan Ketidakadilan Pelayanan Publik Di Kantor Dinas Kependudukan Dan Pencatatan Sipil Kota Pekanbaru,” *Jurnal el Riyash (jel)*, Vol. 8, No. 2, P. 77, Dec. 2017, Doi: 10.24014/Jel.V8i2.4809.
- [3] H. P. Mendrofa, B. Supriadi, S. H. Yustiari, And A. F. Lubis, “Analisis Efektivitas Reformasi Birokrasi Dalam Meningkatkan Kualitas Pelayanan Publik Di Indonesia,” Vol. 8, No. 1, 2025, Doi: 10.56338/Jks.V8i1.6811.
- [4] B. Rudianto And Y. E. Achyani, “Penerapan Metode *Rapid Application Development* Pada Sistem Informasi Persediaan Barang Berbasis Web,” *Bianglala informatika (bi)*, Vol. 8, No. 2, Pp. 117–122, Sep. 2020, Doi: 10.31294/Bi.V8i2.8930.
- [5] Ronald J. Leach, *Introduction To Software Engineering*, Second 2nd. Boca Raton, Fl, Usa: Crc Press, 2016.
- [6] P. Beynon-Davies, *Business Information Systems, Third Edition*. London: Red Globe Press, 2020.
- [7] S. McConnell, *Rapid Development: Taming Wild Software Schedules*, 25. Print., Vol. 1, 3 Vols. Redmond, Wash: Microsoft Press, 2016.
- [8] Nurman Hidayat And Kusuma Hati, “Penerapan Metode *Rapid Application Development* (Rad) Dalam Rancang Bangun Sistem Informasi Rapor *Online* (Siraline),” *jurnal sistem informasi (Jsi)*, Vol. 10, No. 1, Pp. 8–17, Feb. 2021, Doi: 10.51998/Jsi.V10i1.352.
- [9] T. Yunita, Keukeu Saepul Rahayu, Marshella Zalia Putri Setyadi, Nira Ayuhana Nurlitha, And Intan Sulistyaningrum Sakkinah, “Optimalisasi Pengelolaan Transaksi Dengan Merancang Aplikasi *E-Payment* Untuk Umkm Basrenk.Id,” *Prosiding Seminar Nasional Teknologi dan Sistem Informasi Sitasi*, Vol. 3, No. 1, Pp. 667–676, Nov. 2023, Doi: 10.33005/Sitasi.V3i1.676.

- [10] Paul C. Jorgensen, *Software Testing*, Fourth. Newyork: Crc Press Taylor Francis Group, 2014. Accessed: Mar. 16, 2025. [Online]. Available: [Http://Www.Taylorandfrancis.Com](http://www.taylorandfrancis.com)
- [11] E. Suprpto, “User Acceptance Testing (UAT) Refreshment Pbx Outlet Site Bni Kanwil Padang,” *Jurnal Civronlit Unbari Civronlit*, Vol. 6, No. 2, P. 54, Oct. 2021, Doi: 10.33087/Civronlit.V6i2.85.
- [12] John Inc., *Visual Studio Code End To End Editing And Debugging*, 1st Ed., Vol. 1, 1 Vols. Indianapolis, In: John Wiley & Sons, Inc., 2019 di akses pada 1 Maret 2025.
- [13] Scott Chacon And Ben Straub, *Progit Everything You Need Toknow About Git*, 2nd Ed., Vol. 1, 2 Vols. Online/Open Source: Git.Scm, 2025 di akses pada 1 Maret 2025.
- [14] R. Nixon, *Learning Php, Mysql & Javascript: With JQuery, Css & Html5*, Fourth Edition. Sebastopol, Ca: O’reilly Media, Inc, 2014.
- [15] A. Dennis, B. H. Wixom, And R. M. Roth, *System Analys And Design*, Fifth 5th. Hoboken, Nj, Usa: Hoboken, 2012.
- [16] M. Stauffer, *Laravel: Up & Running*, 3rd Ed. Sebastopol, Ca, Usa: O’reilly Media, 2023.
- [17] G. Paolone, R. Paesani, M. Marinelli, And P. Di Felice, “Empirical Assessment Of The Quality Of Mvc Web Applications Returned By Xgenerator,” *Computers*, Vol. 10, No. 2, P. 20, Feb. 2021, Doi: 10.3390/Computers10020020.
- [18] O. Suria, “A Statistical Analysis Of System Usability Scale (Sus) Evaluations In Online Learning Platform,” *Journalisi*, Vol. 6, No. 2, Pp. 992–1007, Jun. 2024, Doi: 10.51519/Journalisi.V6i2.750.
- [19] N. Kumar Miryala, “Evolving Trends In Open-Source Rdbms: Performance, Scalability And Security Insights,” *International Journal of Science and Research (IJSR)*, Vol. 13, No. 2, Pp. 494–500, Feb. 2024, Doi: 10.21275/Sr24126224648.
- [20] S. Soni, P. Chandra, A. Gupta, D. K. Netam, S. Kumar, And K. Tiwary, “Design And Implementation Of Inventory Management System For University,” *European Chemical Bulletin*, Vol. 12, No. 4, Pp. 13746–13759, 2023, Doi: 10.3307.
- [21] Y. Fitrisia, M. Fadhli, And J. Abeth Teofilus Pasaribu, “Rancang Bangun Sistem Informasi Pengelolaan Inventory Berbasis Website Menggunakan Metode

- Rop Dan Eoq (Studi Kasus: Grosir Pa1000 Raja),” *Jurnal Komputer Terapan Jkt*, Vol. 8, No. 2, Pp. 368–380, Dec. 2022, Doi: 10.35143/Jkt.V8i2.5774.
- [22] K. V. Jeeva Padmini, I. Perera, And H. M. N. Dilum Bandara, “*Applying Agile Practices To Avoid Chaos In User Acceptance Testing: A Case Study*,” In *2016 Moratuwa Engineering Research Conference (Mercon)*, Moratuwa, Sri Lanka: Ieee, Apr. 2016, Pp. 96–101. Doi: 10.1109/Mercon.2016.7480122.
- [23] G. W. Sasmito And D. S. Wibowo, “*Implementation Of Rapid Application Development Method In The Development Of Geographic Information Systems Of Industrial Centers*,” *Journal Of Information And Communication Convergence Engineering*, Vol. 18, No. 3, Pp. 194–200, 2020, Doi: 10.6109/Jicce.2020.18.3.194.
- [24] F. Asrin, “Penguujian Fungsionalitas Sistem Inventaris Barang Pada Sekolah Menengah Kejuruan Citra Borneo Menggunakan *Black Box Testing*,” *Ilkominfo*, Vol. 6, No. 2, Pp. 131–143, Jul. 2023, Doi: 10.47324/Ilkominfo.V6i2.202.
- [25] A. Fadli, M. I. Zulfa, A. W. Widhi Nugraha, A. Taryana, And M. S. Aliim, “Analisis Perbandingan Unjuk Kerja *Database SQL* Dan *Database NoSQL* Untuk Mendukung Era Big Data,” *Jurnal Nasional Teknik Elektro Jnte*, Vol. 9, No. 3, Nov. 2020, Doi: 10.25077/Jnte.V9n3.774.2020.
- [26] A. S. Putra, “Sistem Manajemen Pelayanan Pelanggan Menggunakan PHP Dan MySQL (Studi Kasus Pada Toko Surya),” *Tekinfor*, Vol. 22, No. 1, Pp. 100–116, Jun. 2021, Doi: 10.37817/Tekinfor.V22i1.1190.
- [27] Teknik Informatika Universitas Khairun And A. Mubarak, “Rancang Bangun Aplikasi Web Sekolah Menggunakan UML (*Unified Modeling Language*) Dan Bahasa Pemrograman PHP (*PHP Hypertext Preprocessor*) Berorientasi Objek,” *Jiko*, Vol. 2, No. 1, Pp. 19–25, Apr. 2019, Doi: 10.33387/Jiko.V2i1.1052.
- [28] M. F. Fadilah, N. Rahaningsih, And R. D. Dana, “Evaluasi Usabilitas Sistem Menggunakan Metode *System Usability Scale* (SUS) Pada Aplikasi Akhlaqu Dengan Penerapan Teknik Indexing MongoDB,” *Jurnal Sistem Informasi dan Informatika (Simika)* Vol. 7, No. 1, P. 14, 2024, Doi: 10.47080/Simika.V7i1.3070.
- [29] D. P. Kesuma, “Penggunaan Metode *System Usability Scale* Untuk Mengukur Aspek *Usability* Pada Media Pembelajaran Daring Di Universitas Xyz,” *Jatissi*, Vol. 8, No. 3, Pp. 1615–1626, Sep. 2021, Doi: 10.35957/Jatissi.V8i3.1356.

- [30] Aliyah Aliyah, Nahrin Hartono, And Asrul Azhari Muin, “Penggunaan *User Acceptance Testing (Uat)* Pada Pengujian Sistem Informasi Pengelolaan Keuangan Dan Inventaris Barang,” *Jurnal Sains dan Teknologi Informasi Switch*, Vol. 3, No. 1, Pp. 84–100, Dec. 2024, Doi: 10.62951/Switch.V3i1.330.
- [31] I. Afrianto, A. Heryandi, A. Finadhita, And S. Atin, “*User Acceptance Test For Digital Signature Application In Academic Domain To Support The Covid-19 Work From Home Program*,” *STIKOM Tunas Bangsa* Vol. 5, No. 36, 2021.