



Mysql

Bootstrap

Ajax

Laravel



PROYEK SISTEM INFORMASI
EXPLORA
ESA GALLERY, CIAMIS

Pramudya

Rina Fitri Rusli

Wida Aryanti

Deddy Supriadi, M.Kom

Iqbal Dzulfiqar Iskandar, S.T,M.Kom

2024

KATA PENGANTAR

Puji dan syukur kehadirat Allah SWT, yang telah melimpahkan rahmat dan karunia-Nya, sehingga pada akhirnya penyusun dapat menyelesaikan tugas ini dengan baik. Tuga Akhir pada Program Diploma Tiga ini penulis sajikan dalam bentuk buku yang sederhana. Adapun judul Tugas Akhir yang penyusun ambil sebagai berikut, “**Sistem Informasi Esa Gallery (Explora)**”.

Tujuan penulisan Tugas Akhir pada Program Diploma Tiga ini dibuat sebagai salah satu syarat kelulusan Program Diploma Universitas Bina Sarana Informatika. Sebagai bahan penulisan diambil berdasarkan hasil penelitian (eksperimen), observasi dan beberapa sumber literatur yang mendukung penulisan ini. Penyusun menyadari bahwa tanpa bimbingan dan dorongan dari semua pihak, maka penulisan Tugas Akhir ini tidak akan berjalan lancar. Oleh karena itu pada kesempatan ini, ijinkanlah penulis menyampaikan ucapan terima kasih kepada :

1. Rektor Universitas Bina Sarana Informatika.
2. Dekan Fakultas Teknik dan Informatika Universitas Bina Sarana Informatika.
3. Ketua Program Studi Sistem Informasi Universitas Bina Sarana Informatika Kota Tasikmalaya.
4. Seluruh staff pengajar fakultas teknik dan informatika Universitas Bina Sarana Informatika yang telah memberikan ilmu pengetahuan yang tak ternilai selama penulis menempuh pendidikan di Universitas Bina Sarana Informatika Kota Tasikmalaya.
5. Evi Sophian Affandi Selaku Pemilik Perusahaan Esa Gallery.
6. Orang tua tercinta yang telah memberikan dukungan moral maupun spiritual.
7. Rekan-rekan mahasiswa kelas SI-6A dan SI-6B.

Serta semua pihak yang terlalu banyak untuk disebut satu persatu sehingga terwujudnya penulisan ini. Penyusun menyadari penulisan Tugas Akhir ini masih jauh sekali dari sempurna, untuk itu penyusun mohon kritik dan saran yang bersifat membangun demi kesempurnaan penulisan di masa yang akan datang.

Akhir Kata semoga Tugas Akhir ini dapat berguna bagi penulis khususnya dan bagi para pembaca yang berminat pada umumnya.

Tasikmalaya, Juli 2024

Penyusun



DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	iii
BAB I	1
PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Tools Pengembangan Aplikasi	4
1.3 Deskripsi Aplikasi	12
BAB II	13
PERANCANGAN APLIKASI	13
2.1 Rancangan Sistem	13
2.2 Rancangan Database.....	55
2.3 Rancangan Form Master.....	70
2.4 Rancangan Transaksi.....	144
2.5 Rancangan Laporan	161
2.6 Pengujian Unit.....	187
BAB III	218
KESIMPULAN	218
3.1 Kesimpulan.....	218
3.2 Saran	219
DAFTAR PUSTAKA	220
BIOGRAFI PENULIS	223

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Menurut Undang-Undang Pasal 1 No 1 dan 2 Tahun 2017, “Jasa Konstruksi adalah layanan jasa konsultansi konstruksi dan/atau pekerjaan konstruksi”, “Konsultansi konstruksi adalah layanan keseluruhan atau sebagian kegiatan yang meliputi pengkajian, perencanaan, perancangan, pengawasan, dan manajemen penyelenggaraan konstruksi suatu bangunan”. Menurut Peraturan Pemerintah Nomor 22 tahun 2020 tentang Peraturan Pelaksanaan Undang-Undang Nomor 2 tahun 2017 tentang Jasa Konstruksi, “Konstruksi adalah rangkaian kegiatan untuk mewujudkan, memelihara, menghancurkan bangunan yang Sebagian atau seluruhnya menyatu dengan tanah atau tempat kedudukannya menyatu dengan tanah, bangunan konstruksi adalah wujud fisik hasil jasa konstruksi”. “Pekerjaan konstruksi adalah keseluruhan atau sebagian kegiatan yang meliputi pembangunan, pengoperasian, pemeliharaan, pembongkaran, dan pembangunan kembali suatu bangunan.” (Undang-Undang Pasal 1 Nomor 3 Tahun 2017 Tentang Jasa Konstruksi).

Berkaitan dengan bidang jasa konstruksi, Esa Gallery merupakan perusahaan swasta yang bergerak di bidang jasa, seperti jasa konstruksi dan bangunan, interior dan exterior, taman, tebing, furnitur, pembuatan kolam renang, dan kolam koi. Dari hasil riset yang dilakukan di Esa Gallery dalam lingkup sistem manajemen proyek di perusahaan ini. Telah teridentifikasi pada sistem berjalan, yaitu manajemen proyek yang dilakukan belum terkomputerisasi. Secara khusus, sistem dokumentasi proyek dan platform untuk memamerkan karya-karya dari perusahaan ini belum ada.

Saat ini, Esa Gallery menggunakan metode manual untuk mengelola proyek mereka. Dokumen hasil proyek, laporan rincian anggaran dan desain saat ini disimpan

dalam bentuk fisik atau tersebar di berbagai lokasi digital yang tidak terpusat, seperti komputer pribadi atau email, sehingga meningkatkan risiko kehilangan atau kerusakan. Selain itu, foto-foto dan media lain yang dihasilkan selama proyek disimpan di perangkat individu atau dalam penyimpanan digital yang tidak terstruktur, tanpa adanya sistem pengarsipan yang baik, membuat media ini rentan terhadap kehilangan atau terhapus secara tidak sengaja. Karena adanya permasalahan tersebut, peneliti bermaksud untuk menghadirkan (Explora) sebagai solusi untuk mengelola proyek galeri perusahaan dengan lebih baik, memastikan semua dokumen dan media tersimpan secara terpusat dan terorganisir, serta meningkatkan efisiensi dan keamanan dalam pengelolaan proyek.

Metode yang akan diterapkan pada riset ini menggunakan pendekatan manajemen proyek berbasis Scrum karena fleksibilitas dan kemampuan adaptasinya terhadap perubahan yang tinggi, yang sangat cocok untuk lingkungan proyek galeri dan pameran yang dinamis. Scrum memungkinkan pengelolaan tugas dalam sprint pendek yang terfokus, serta peningkatan terus-menerus melalui evaluasi rutin (Faiz & Idris, 2022). Selain itu, teknik observasi yang digunakan dalam riset ini akan menggunakan metode observasi partisipatif. Dalam metode ini, peneliti akan terlibat langsung dalam lingkungan kerja untuk mengamati secara mendalam proses bisnis dan interaksi antar anggota tim. Observasi partisipatif memungkinkan peneliti untuk mendapatkan wawasan yang lebih akurat dan rinci tentang kebutuhan pengguna serta tantangan yang dihadapi selama pengembangan sistem informasi.

Dalam penelitian (Muzadi & Chotijah, 2023) membahas pemanfaatan teknologi informasi dan manajemen pembelajaran pada masa pandemi di Indonesia. Penelitian (Muhammad Pauhrizi et al., 2022) membahas perspektif kontruksi itu penting dalam penataan artistik, dan menekankan penggunaan Teknik visual dan desain dalam proyek

seni. Penelitian (Budiman et al., 2024) meningkatkan manajemen proyek pengelolaan melalui sistem informasi di industri konstruksi dan desain untuk meningkatkan efisiensi dan menurunkan biaya. Berdasarkan hal tersebut, ada beberapa kesenjangan atau terdapat kekurangan pada temuan sebelumnya, sehingga dalam proses pengembangan aplikasi yang kami lakukan agar lebih dsempurnakan. Hasil temuan sebelumnya terdapat kekurangan pada segi kelemahan dari penelitian-penelitian ini adalah kurangnya fokus pada manajemen media visual dan dokumen dalam konteks seni dan pameran, serta tidak membahas pengarsipan digital secara terstruktur untuk proyek-proyek seni.

Maka dari itu pada riset ini diusulkan perancangan galeri perusahaan kontraktor dengan objek instrument penelitian yang berbeda yaitu Esa Gallery dengan tambahan fitur galeri seni sebagai penyempurna dari riset terdahulu. Penelitian ini bertujuan untuk menyempurnakan ini dengan mengembangkan sistem manajemen proyek yang tidak hanya efisien dalam mengelola dokumen dan media visual tetapi juga terstruktur dan terpusat. Dengan demikian, kami akan menghadirkan solusi yang lebih komprehensif yang mencakup semua aspek yang diperlukan untuk manajemen proyek seni secara digital, meminimalkan risiko kehilangan data, dan memastikan aksesibilitas serta pengarsipan yang baik.

Melalui pengembangan sistem informasi ini dapat membantu untuk pemasaran, meningkatkan daya saing industri, serta memberikan platform yang kuat bagi Perusahaan, secara teknis sistem harus stabil, cepat dan aman serta mampu mendukung tujuan Perusahaan dalam memasarkan produk, meningkatkan keunggulan kompetitif dan berkontribusi pada pertumbuhan bisnis.

Berdasarkan hasil riset di Esa Gallery, ditemukan bahwa sistem manajemen proyek mereka saat ini menggunakan metode manual, menyebabkan masalah dalam

pengelolaan dokumen dan media visual secara terpusat dan terstruktur. Metode observasi partisipatif dan pendekatan manajemen proyek berbasis Scrum untuk memperbaiki masalah tersebut, dengan tujuan menyempurnakan sistem manajemen proyek dengan tambahan fitur galeri seni. Ini akan memastikan efisiensi dalam mengelola proyek seni secara digital, meminimalkan risiko kehilangan data, dan meningkatkan aksesibilitas serta pengarsipan yang baik dan solusi yang dikembangkan dapat membantu dalam pemasaran dan meningkatkan daya saing industri bagi Perusahaan Esa Gallery.

1.2 Tools Pengembangan Aplikasi

A. Aplikasi

Menurut (Muhammad1, 2023) Kata "aplikasi" berasal dari kata "aplikasi", yang berarti "penerapan, lamaran, atau penggunaan." Program siap pakai yang dapat digunakan untuk mencapai tujuan tertentu disebut aplikasi. "aplikasi" berasal dari kata "aplikasi", dan "aplikasi" biasanya mengacu pada pemrosesan data yang diinginkan, atau komputasi yang diinginkan.

Menurut ((David et al., 2024) Aplikasi adalah program yang siap untuk digunakan yang dirancang untuk melakukan fungsi tertentu bagi pengguna yang menggunakan layanan aplikasi, serta aplikasi lain yang dapat digunakan untuk mencapai tujuan tertentu.

Menurut Aplikasi (Sri et al., 2024) adalah sistem yang dirancang untuk mengelola data menggunakan aturan atau tata letak dari bahasa pemrograman tertentu. Program komputer yang dirancang untuk melakukan tugas-tugas tertentu disesuaikan dengan kebutuhan pengguna disebut aplikasi.

B. Sistem

Menurut (Wismoaji, 2024) Sistem ini berasal dari sistem Latin dan sistema Yunani. Pengertian sistem adalah sekumpulan komponen atau elemen yang dihubungkan untuk memudahkan aliran informasi, materi, atau energi untuk mencapai suatu tujuan. Pengertian sistem juga dapat didefinisikan sebagai sekumpulan unsur atau elemen dalam sistem yang bekerja sama untuk menjalankan proses atau fungsi tertentu untuk mencapai tujuan sistem.

C. Sistem Informasi

Menurut (Purnama, 2024) Sistem Informasi adalah sistem di dalam suatu organisasi yang menggabungkan kebutuhan pengolahan transaksi harian, mendukung operasi, memenuhi fungsi manajemen dan aktivitas strategis, dan memberikan laporan kepada pihak luar. Sistem informasi juga dapat didefinisikan sebagai kerangka kerja yang mengatur sumber daya (manusia dan komputer) untuk mengubah input (masukan) menjadi keluaran (informasi) untuk mencapai tujuan bisnis.

Menurut (Reza M Rahim Nur, 2024) Sistem Informasi adalah rangkaian sistem yang terorganisir dalam suatu organisasi. Sistem informasi terdiri dari sekumpulan komponen, baik manual maupun berbasis komputer, yang dirancang untuk menghimpun dan menyiapkan data yang mengandung informasi keluaran untuk pemakai, atau sekumpulan perangkat keras dan perangkat lunak yang terhubung untuk menghasilkan dan memproses data menjadi informasi yang berguna.

D. Kontraktor

Menurut (Mita Qurrota Ayunin, 2023) kontraktor juga disebut sebagai pelaksana konstruksi, adalah penyedia jasa yang memiliki keahlian khusus dalam melaksanakan pekerjaan konstruksi dan memiliki kemampuan untuk mengkoordinasikan operasinya untuk mencapai suatu tujuan. planing menjadi struktur atau bentuk fisik lainnya. Kontraktor adalah perusahaan atau individu yang bekerja dengan pihak lain untuk melakukan konstruksi atau proyek pembangunan. Mereka bertanggung jawab atas pelaksanaan proyek konstruksi, koordinasi tenaga kerja, pengaturan pengadaan bahan dan peralatan, dan memastikan bahwa proyek selesai sesuai dengan spesifikasi, jadwal, dan anggaran yang telah disepakati.

E. Bahasa Pemograman

Menurut (Fathonia & Yahfizham, 2024) Bahasa pemrograman, juga disebut sebagai "bahasa komputer", adalah instruksi standar untuk memerintahkan komputer. Untuk seorang programmer, penggunaan bahasa ini sangat membantu dalam mengolah data yang akan dikerjakan dan disimpan pada komputer serta dalam menyelesaikan berbagai situasi.

F. Website

Menurut (Sri et al., 2024) Website adalah kumpulan elemen yang terdiri dari teks, gambar, dan suara yang disediakan melalui jaringan internet dan dapat diakses oleh setiap orang yang terhubung ke internet. Secara terminologi, website adalah kumpulan halaman web, yang biasanya terkumpul dalam domain atau subdomain di World Wide Web (WWW) di internet. Halaman web adalah

dokumen yang ditulis dalam format HTML (Hyper Text Markup Language) dan hampir selalu dapat diakses melalui protokol HTTP. Protokol ini mengirimkan informasi dari server website kepada pengguna melalui web browser.

G. Database

Menurut (Syahputri et al., 2023) Database adalah koleksi data yang terstruktur yang disimpan dalam media penyimpanan komputer dan dapat diakses dan dikelola menggunakan perangkat lunak khusus. Tujuan database adalah untuk menyimpan, mengelola, dan mengorganisasi data sehingga orang dapat dengan mudah, aman, dan terstruktur mengakses data mereka.

H. HTML (*Hyper Text Markup Languange*)

Menurut (Purnama, 2024) HTML adalah bahasa pemrograman yang digunakan untuk menampilkan dokumen dalam browser. Banyak tag dan atribut yang digunakan oleh HTML digunakan untuk menentukan struktur dokumen web dan tata letak tampilan. Dokumen HTML dimulai dengan tag <HTML> dan diakhiri dengan tag </HTML>.

Menurut (Arif Nur Rahman, 2024) Bahasa HTML adalah singkatan dari HyperText Markup Language. Istilah "hypertext" digunakan karena dalam HTML, teks biasa dapat melakukan fungsi tambahan, seperti mengubahnya menjadi tautan yang memungkinkan pengguna berpindah dari satu halaman ke halaman lain hanya dengan mengklik teks tersebut. Namun, dalam praktiknya, HTML tidak hanya memiliki teks yang dapat menjadi tautan. Karena bahasa HTML menggunakan tanda (mark) untuk menandai bagian teks, istilah "Markup Language" digunakan.

I. CSS (*Cascading Style Sheet*)

Menurut (Sidiq Amroni1, 2024) Cascading Style Sheets (CSS) adalah bahasa pemrograman desain web yang memiliki kemampuan untuk mengatur sistem identifikasi tampilan halaman web yang dicatat melalui bahasa indikator (markup language).

J. PHP (*Hypertext Preprocessor*)

Menurut (Sri et al., 2024) PHP, singkatan dari Perl Hypertext Preprocessor, adalah kode atau skrip yang dilakukan di server, dan hasilnya dikirimkan ke browser. Bahasa pemrograman PHP bahasa pemrograman yang digunakan pada server web.

K. PHPMyAdmin

Menurut (Gumilang et al., 2023) PHPMyAdmin ialah sebuah web server yang digunakan untuk mengelola database. Ini digunakan untuk mengelola program web yang sudah dirancang yang membutuhkan database untuk berfungsi secara sinkron. MySQL adalah program yang digunakan untuk membuat query, membuat database, dan mengedit data dengan RDBMS, perangkat lunak sistem yang menjalankan fungsi pengolahan data.

L. Javascript

Menurut (Arif Nur Rahman, 2024) JavaScript adalah bahasa pemrograman yang digunakan untuk membuat dan mengontrol konten website agar tetap dinamis. Contoh konten situs yang dinamis termasuk komponen yang dapat

bergerak atau berubah tanpa membuat pengguna harus mengubah halaman situs secara manual.

M. Mysql

Menurut (Sri et al., 2024) Salah satu aplikasi database adalah MySQL, yang dapat digunakan untuk menyimpan data dalam aplikasi. SQL adalah bahasa komputer yang memenuhi standar American National Standard Institute (ANSI) digunakan untuk mengakses dan melakukan manipulasi sistem database. Anda dapat melakukan administrasi dalam basis data MySQL dengan modul perintah yang sudah termasuk, yaitu perintah "mysql" dan "mysql admin".

N. Laragon

Menurut (Shahib et al., 2024), Laragon adalah platform pengembangan web yang membuat pengembangan dan pengujian aplikasi web lebih mudah. Dengan menginstalnya pada sistem operasi Windows, pengembang dapat dengan mudah mengatur server web, database, dan komponen pengembangan lainnya tanpa harus melakukan konfigurasi manual yang rumit.

O. Laravel

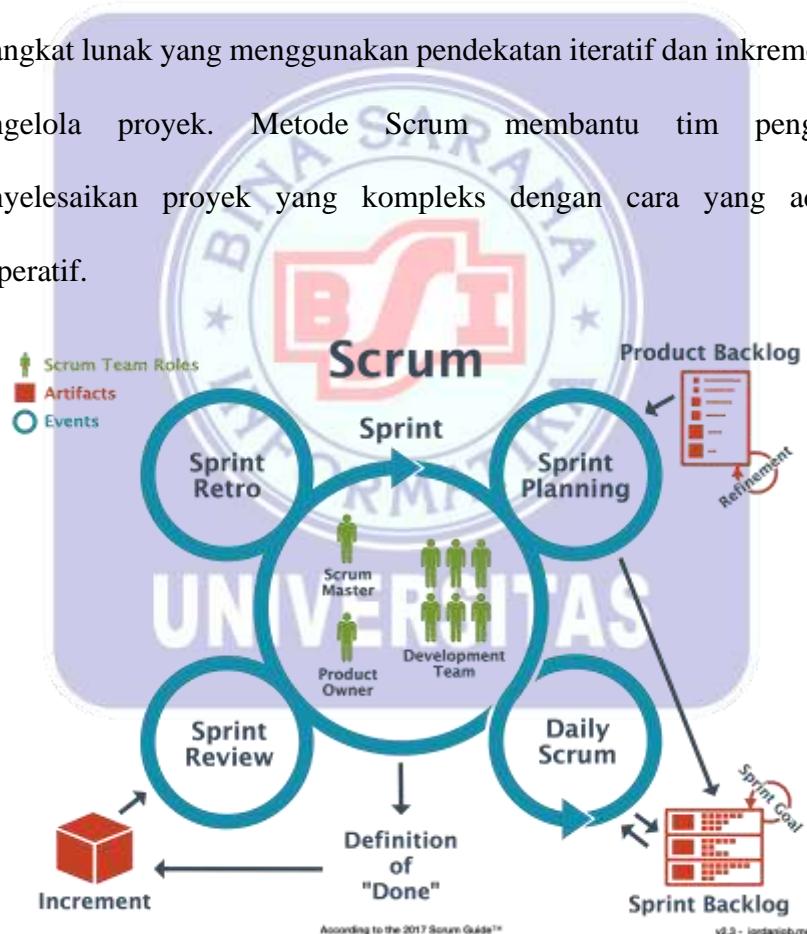
Menurut (Hafsari et al., 2024) Laravel adalah framework PHP yang open source yang menggunakan konsep Model, View, and Controller (MVC) untuk membuat aplikasi berbasis website.

P. Bootstrap

Menurut (Dody Firmansyah, 2023) Bootstrap adalah salah satu kerangka kerja CSS yang digunakan oleh pembangun web untuk membantu mereka mempercepat, mempercepat, dan meningkatkan tampilan situs mereka. Untuk front-end support, atau client-side support, bootstrap ini sangat bermanfaat.

Q. Metode Scrum

Menurut (Giawa & Abdy, 2024) Scrum adalah kerangka kerja pengembangan perangkat lunak yang menggunakan pendekatan iteratif dan inkremental untuk mengelola proyek. Metode Scrum membantu tim pengembangan menyelesaikan proyek yang kompleks dengan cara yang adaptif dan kooperatif.



Gambar I.1 Representasi Dari Metode Scrum

Gambar I.1 merupakan representasi dari metode scrum, penjelasannya sebagai berikut. Dalam metode Scrum, pengembangan perangkat lunak dibagi menjadi

serangkaian iterasi pendek yang disebut “Sprint”. Setiap Sprint memiliki waktu terbatas, biasanya antara satu hingga empat minggu, dan memiliki tujuan yang jelas yang harus dicapai. Tim Scrum terdiri dari tiga peran utama: Product Owner, Scrum Master, dan Tim Pengembang.

1. Tim Scrum : terdiri dari tiga peran utama

- a. Product Owner (Pemilik Produk): Bertanggung jawab atas backlog produk dan memastikan tim mengerjakan tugas-tugas yang memberikan nilai maksimal.
- b. Scrum Master: Bertanggung jawab memastikan Scrum diimplementasikan dengan benar, membantu mengatasi hambatan, dan melindungi tim dari gangguan eksternal.
- c. Tim Pengembang: Kelompok multifungsi yang mengerjakan tugas-tugas yang telah diprioritaskan oleh Product Owner untuk menghasilkan peningkatan produk.

2. Artefak Scrum:

- a. Product Backlog: Daftar tugas yang perlu diselesaikan untuk proyek, diprioritaskan oleh Product Owner.
- b. Sprint Backlog: Subset dari product backlog yang akan dikerjakan selama sprint tertentu.
- c. Increment (Inkrementasi): Hasil kerja dari sprint yang bisa dipakai dan diuji.

3. Acara dalam scrum:

- a. Sprint: Periode waktu tetap di mana pekerjaan dilakukan untuk membuat peningkatan produk.

- b. Sprint Planning (Perencanaan Sprint): Pertemuan di awal sprint di mana tim menentukan tujuan sprint dan memilih item dari product backlog untuk dikerjakan.
- c. Daily Scrum (Scrum Harian): Pertemuan singkat harian (15 menit) di mana tim mengevaluasi kemajuan dan mengidentifikasi hambatan.
- d. Sprint Review: Pertemuan di akhir sprint untuk mendemonstrasikan apa yang telah dicapai dan mendapatkan masukan dari pemangku kepentingan.
- e. Sprint Retrospective (Retrospektif Sprint): Pertemuan setelah sprint review di mana tim merefleksikan proses kerja dan mengidentifikasi area untuk perbaikan.

1.3 Deskripsi Aplikasi

Explora adalah aplikasi sistem informasi terintegrasi yang dikembangkan khusus untuk Esa Gallery, sebuah perusahaan yang bergerak dalam bidang kontraktor, desain interior, dan furniture. Aplikasi ini dirancang untuk memudahkan pengelolaan galeri proyek konstruksi, desain interior, dan furniture, serta meningkatkan koordinasi antara tim, transparansi informasi, dan kepuasan pelanggan.

BAB II

PERANCANGAN APLIKASI

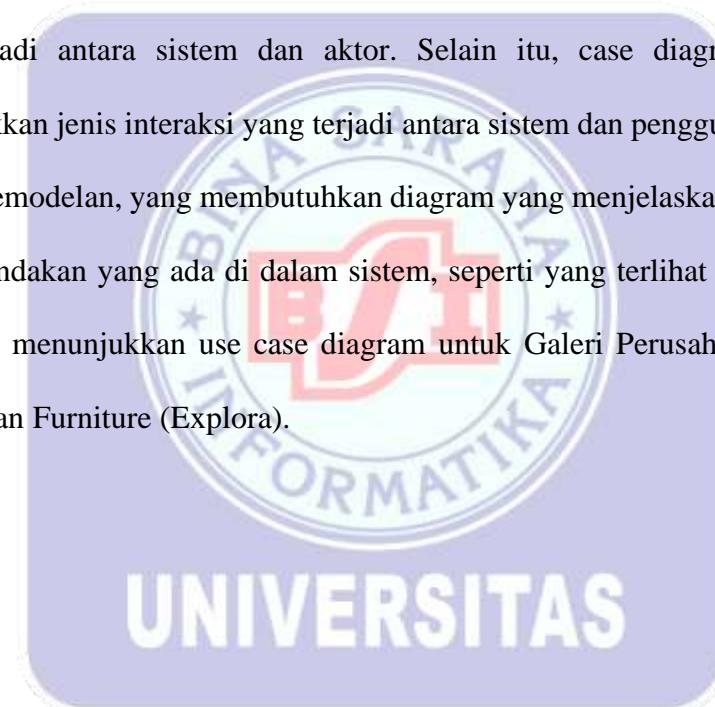
2.1 Rancangan Sistem

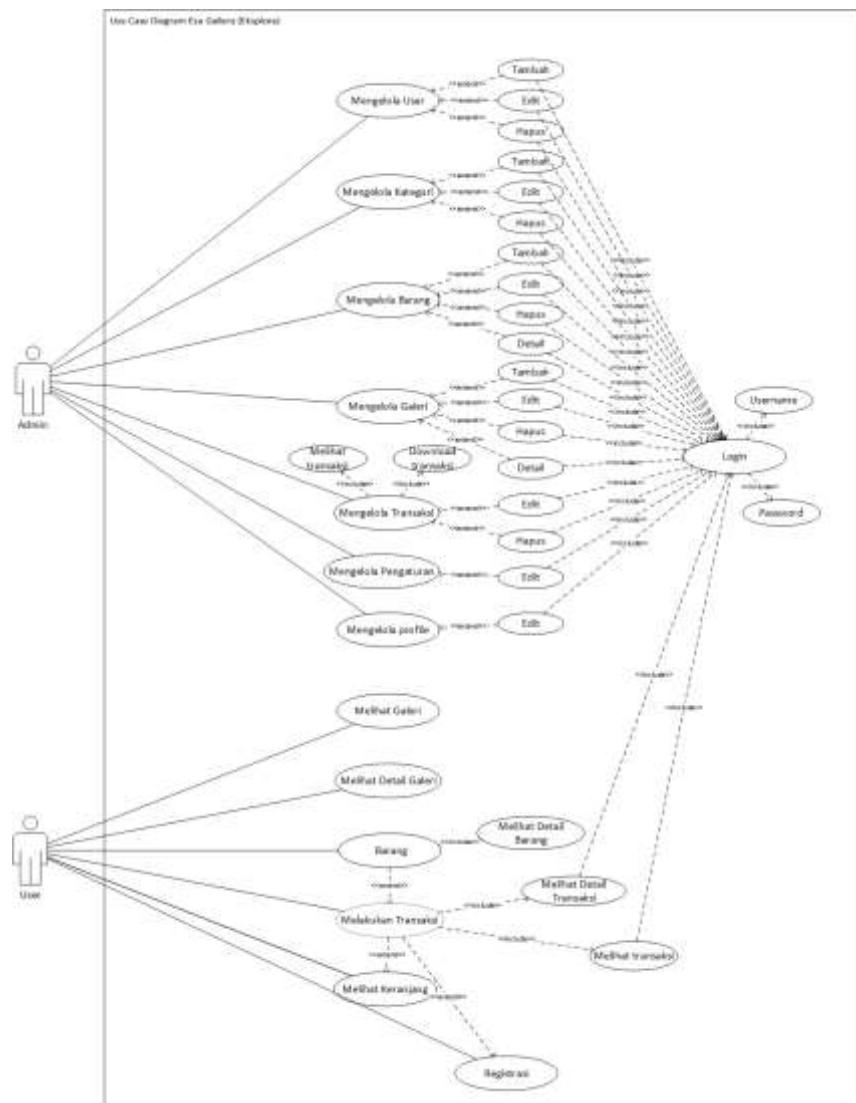
“UML adalah sekumpulan alat yang biasanya digunakan untuk abstraksi sistem atau perangkat lunak berbasis objek. (Noviantoro et al., 2022)“

A. Use Case Diagram

Menurut (Ega Hansel Sanjaya1), 2024) Salah satu jenis diagram Unified Modelling Language (UML) adalah case diagram, yang menunjukkan hubungan yang terjadi antara sistem dan aktor. Selain itu, case diagram juga dapat menunjukkan jenis interaksi yang terjadi antara sistem dan pengguna. Dimulai dengan pemodelan, yang membutuhkan diagram yang menjelaskan tindakan aktor dengan tindakan yang ada di dalam sistem, seperti yang terlihat dalam use case.

Tabel 2.1 menunjukkan use case diagram untuk Galeri Perusahaan Kontraktor, Interior dan Furniture (Explora).





Gambar II.1
Gambar II. 1 Usecase Esa Gallery (Explora)

Sumber : Hasil Penelitian (2024)

1. Login

Tabel II. 1

Deskripsi Use Case Login

Use Case Name	Login
Requirements	User dapat mengakses fitur
Goal	User menggunakan fitur sesuai dengan role secara online melalui web
Pre-conditions	User tidak melakukan login
Post-conditions	User melakukan login agar dapat mengakses fitur web
Failed end condition	User tidak dapat login dan mengakses fitur web
Primary Actors	Admin, User
Main Flow / Basic Path	<ol style="list-style-type: none">1. User mengakses halaman login pada web2. User mengisi email dan password3. User mengeklik tombol Login4. Sistem memverifikasi data user5. Sistem menampilkan pesan “Login berhasil”6. Sistem melanjutkan ke halaman beranda
Alternative Flow	Sistem melakukan verifikasi data yang sesuai dengan data user yang ada pada database
Invariant B	<ol style="list-style-type: none">1. User mengakses halaman login pada web2. User mengisi data email dan password3. User mengklik tombol login4. Sistem memverifikasi data user5. Sistem tidak menemukan data user6. Sistem menampilkan pesan “Email atau password tidak valid”

Sumber : Hasil Penelitian (2024)

2. Mengelola Kategori

Tabel II. 2
Deskripsi Use Case Mengelola Kategori

Use case name	Mengelola kategori
Requirement	User dapat menambah, mengedit dan menghapus kategori
Goal	User menggunakan fitur tambah, edit dan hapus pada mengelola kategori
Pre-conditions	User telah melakukan login
Post-conditions	User mengakses halaman kategori
Failed end condition	User tidak dapat mengakses halaman kategori
Primary Actor	Admin
Main Flow / Basic Path	<p>a. User melakukan tambah data kategori</p> <ol style="list-style-type: none">1) User mengakses halaman web admin kategori2) User mengklik tombol tambah3) Sistem menampilkan modal form untuk mengisi nama4) User mengisi data form nama5) User mengklik tombol simpan6) Sistem memverifikasi data7) Sistem menyimpan data kategori8) Sistem menampilkan pesan “Data kategori ditambahkan”

	<p>9) Sistem menampilkan data yang disimpan berbentuk tabel</p> <p>b. User melakukan edit data kategori</p> <ol style="list-style-type: none"> 1) User mengakses halaman web admin kategori 2) User mengklik tombol edit yang dipilih 3) Sistem menampilkan data pada modal form untuk mengubah nama 4) User mengubah data form nama 5) User mengklik tombol simpan 6) Sistem memverifikasi data 7) Sistem mengubah data kategori 8) Sistem menampilkan pesan “Data kategori diubah” 9) Sistem menampilkan data yang diubah berbentuk tabel <p>c. User melakukan hapus data kategori</p> <ol style="list-style-type: none"> 1) User mengakses halaman web admin kategori 2) User mengklik tombol hapus yang dipilih 3) Sistem menampilkan dialog konfirmasi dengan memiliki pesan “Apakah Kamu Yakin? ingin menghapus data ini!” 4) User mengklik ya 5) Sistem memverifikasi data 6) Sistem menghapus data kategori
--	--

	<p>7) Sistem menampilkan pesan “Data kategori dihapus”</p> <p>8) Sistem menghapus data yang ada pada tabel</p>
Altenative Flow	<p>a. Sistem dapat melakukan tambah data pada kategori</p> <p>b. Sistem dapat melakukan edit data pada kategori</p> <p>c. Sistem dapat melakukan hapus data pada kategori</p>
Invariant B	<p>a. User melakukan tambah data kategori</p> <p>1) User mengakses halaman web admin kategori</p> <p>2) User mengklik tombol tambah</p> <p>3) Sistem menampilkan modal form untuk mengisi nama</p> <p>4) User mengosongkan data form nama</p> <p>5) User mengklik tombol simpan</p> <p>6) Sistem memberikan validasi pada form yang wajib diisi</p> <p>7) User mengisi ulang form yang kosong</p> <p>8) User mengklik tombol simpan</p> <p>9) Sistem memverifikasi data</p> <p>10) Sistem menyimpan data kategori</p> <p>11) Sistem menampilkan pesan “Data kategori ditambahkan”</p> <p>12) Sistem menampilkan data yang disimpan berbentuk tabel</p>

	<p>b. User melakukan edit data kategori</p> <ol style="list-style-type: none"> 1) User mengakses halaman web admin kategori 2) User mengklik tombol edit yang dipilih 3) Sistem menampilkan data pada modal form untuk mengubah nama 4) User mengosongkan form nama 5) User mengklik tombol simpan 6) Sistem memberikan validasi pada form yang wajib diisi 7) User mengisi ulang form yang kosong 8) User mengklik tombol simpan 9) Sistem memverifikasi data 10) Sistem mengubah data kategori 11) Sistem menampilkan pesan “Data kategori diubah” 12) Sistem menampilkan data yang diubah berbentuk tabel <p>c. User melakukan hapus data kategori</p> <ol style="list-style-type: none"> 1) User mengakses halaman web admin kategori 2) User mengklik tombol hapus yang dipilih 3) Sistem menampilkan dialog konfirmasi dengan memiliki pesan “Apakah Kamu Yakin? ingin menghapus data ini?” 4) User mengklik tidak
--	--

	5) Sistem tidak menghapus data kategori
--	---

Sumber : Hasil Penelitian (2024)

3. Mengelola Barang

Tabel II. 3

Deskripsi Use Case Mengelola Barang

Use case name	Mengelola barang
Requirement	User dapat melakukan tambah, edit, dan hapus pada mengelola barang
Goal	User menambah, mengedit, menghapus dan mengakses data barang
Pre-conditions	User telah melakukan login
Post-conditions	User mengakses halaman barang
Failed end condition	User tidak dapat mengakses halaman barang
Primary Actor	Admin
Main Flow / Basic Path	<p>a. User melakukan tambah data barang</p> <ol style="list-style-type: none"> 1) User mengakses halaman web admin barang 2) User mengklik tombol tambah 3) Sistem menampilkan modal form untuk mengisi nama, kategori, harga dan deskripsi 4) User mengisi data form mengisi nama, kategori, harga dan deskripsi 5) User mengklik tombol simpan 6) Sistem memverifikasi data 7) Sistem menyimpan data barang

	<p>8) Sistem menampilkan pesan “Data barang ditambahkan”</p> <p>9) Sistem menampilkan data yang disimpan berbentuk tabel</p> <p>b. User melakukan edit data barang</p> <ol style="list-style-type: none"> 1) User mengakses halaman web admin barang 2) User mengklik tombol edit yang dipilih 3) Sistem menampilkan data pada modal form untuk mengubah nama, kategori, harga dan deskripsi 4) User mengubah data form nama, kategori, harga dan deskripsi 5) User mengklik tombol simpan 6) Sistem memverifikasi data 7) Sistem mengubah data barang 8) Sistem menampilkan pesan “Data barang diubah” 9) Sistem menampilkan data yang diubah berbentuk tabel <p>c. User melakukan hapus data barang</p> <ol style="list-style-type: none"> 1) User mengakses halaman web admin barang 2) User mengklik tombol hapus yang dipilih 3) Sistem menampilkan dialog konfirmasi dengan memiliki pesan “Apakah Kamu Yakin? ingin menghapus data ini!” 4) User mengklik ya
--	--

	<p>5) Sistem memverifikasi data</p> <p>6) Sistem menghapus data barang</p> <p>7) Sistem menampilkan pesan “Data barang dihapus”</p> <p>8) Sistem menghapus data yang ada pada tabel</p> <p>d. User melakukan tambah pada detail data barang gambar</p> <ol style="list-style-type: none"> 1) User mengakses halaman web admin barang 2) User mengklik tombol detail 3) Sistem menampilkan pada modal form untuk mengisi gambar 4) User mengisi form gambar 5) User mengklik tombol simpan 6) Sistem memverifikasi data 7) Sistem menyimpan data barang gambar 8) Sistem menampilkan pesan “Data barang gambar disimpan” 9) Sistem menampilkan data yang diubah berbentuk tabel <p>e. User melakukan edit pada detail data barang gambar</p> <ol style="list-style-type: none"> 1) User mengakses halaman web admin barang 2) User mengklik tombol edit yang dipilih 3) Sistem menampilkan data pada modal form untuk mengubah gambar 4) User mengubah data form gambar
--	---

	<p>5) User mengklik tombol simpan</p> <p>6) Sistem memverifikasi data</p> <p>7) Sistem mengubah data barang gambar</p> <p>8) Sistem menampilkan pesan “Data barang gambar diubah”</p> <p>9) Sistem menampilkan data yang diubah berbentuk tabel</p> <p>f. User melakukan hapus data barang gambar</p> <p>1) User mengakses halaman web admin barang</p> <p>2) User mengklik tombol hapus yang dipilih</p> <p>3) Sistem menampilkan dialog konfirmasi dengan memiliki pesan “Apakah Kamu Yakin? ingin menghapus data ini?”</p> <p>4) User mengklik ya</p> <p>5) Sistem memverifikasi data</p> <p>6) Sistem menghapus data barang</p> <p>7) Sistem menampilkan pesan “Data barang dihapus”</p> <p>8) Sistem menghapus data yang ada pada tabel</p>
Altenative Flow	<p>a. Sistem dapat melakukan tambah data pada barang</p> <p>b. Sistem dapat melakukan edit data pada barang</p> <p>c. Sistem dapat melakukan hapus data pada barang</p> <p>d. Sistem dapat melakukan tambah data pada barang gambar</p>

	<p>b. Sistem dapat melakukan edit data pada barang gambar</p> <p>c. Sistem dapat melakukan hapus data pada barang gambar</p>
Invariant B	<p>a. User melakukan tambah data</p> <ol style="list-style-type: none"> 1) User mengakses halaman web admin barang 2) User mengklik tombol tambah 3) Sistem menampilkan modal form untuk mengisi nama, kategori, harga dan deskripsi 4) User mengisi data form mengisi nama, kategori, harga dan deskripsi dengan salah satu form ada yang kosong 5) User mengklik tombol simpan 6) Sistem memberikan validasi pada form yang wajib diisi 7) User mengisi ulang form yang kosong 8) User mengklik tombol simpan 9) Sistem memverifikasi data 10) Sistem menyimpan data barang 11) Sistem menampilkan pesan “Data barang ditambahkan” 12) Sistem menampilkan data yang disimpan berbentuk tabel

	<p>b. User melakukan edit data barang</p> <ol style="list-style-type: none"> 1) User mengakses halaman web admin barang 2) User mengklik tombol edit yang dipilih 3) Sistem menampilkan data pada modal form untuk mengubah nama, kategori, harga dan deskripsi 4) User mengubah data form nama, kategori, harga dan deskripsi dengan salah satu form ada yang kosong 5) User mengklik tombol simpan 6) Sistem memberikan validasi pada form yang wajib diisi 7) User mengisi ulang form yang kosong 8) User mengklik tombol simpan 9) Sistem memverifikasi data 10) Sistem mengubah data barang 11) Sistem menampilkan pesan “Data barang diubah” 12) Sistem menampilkan data yang diubah berbentuk tabel <p>c. User melakukan hapus data barang</p> <ol style="list-style-type: none"> 1) User mengakses halaman web admin barang 2) User mengklik tombol hapus yang dipilih 3) Sistem menampilkan dialog konfirmasi dengan memiliki pesan “Apakah Kamu Yakin? ingin menghapus data ini?”
--	--

	<p>4) User mengklik tidak</p> <p>5) Sistem tidak menghapus data barang</p> <p>d. User melakukan tambah pada detail data barang gambar</p> <p>1) User mengakses halaman web admin barang</p> <p>2) User mengklik tombol detail</p> <p>3) Sistem menampilkan pada modal form untuk mengisi gambar</p> <p>4) User mengosongkan form gambar</p> <p>5) User mengklik tombol simpan</p> <p>6) Sistem memberikan validasi pada form yang wajib diisi</p> <p>7) User mengisi ulang form yang kosong</p> <p>8) User mengklik tombol simpan</p> <p>9) Sistem memverifikasi data</p> <p>10) Sistem menyimpan data barang</p> <p>11) Sistem menampilkan pesan “Data barang gambar ditambahkan”</p> <p>12) Sistem menampilkan data yang disimpan berbentuk tabel</p> <p>e. User melakukan edit pada detail data barang gambar</p> <p>1) User mengakses halaman web admin barang</p> <p>2) User mengklik tombol edit yang dipilih</p>
--	---

	<p>3) Sistem menampilkan data pada modal form untuk mengubah gambar</p> <p>4) User mengubah data form gambar</p> <p>5) User mengklik tombol simpan</p> <p>6) Sistem memberikan validasi pada form yang wajib diisi</p> <p>7) User mengisi ulang form yang kosong</p> <p>8) User mengklik tombol simpan</p> <p>9) Sistem memverifikasi data</p> <p>10) Sistem mengubah data barang 11) Sistem menampilkan pesan “Data barang gambar diubah”</p> <p>12) Sistem menampilkan data yang diubah berbentuk tabel</p> <p>f. User melakukan hapus data barang</p> <p>1) User mengakses halaman web admin barang</p> <p>2) User mengklik tombol hapus yang dipilih</p> <p>3) Sistem menampilkan dialog konfirmasi dengan memiliki pesan “Apakah Kamu Yakin? ingin menghapus data ini?”</p> <p>4) User mengklik tidak</p> <p>5) Sistem tidak menghapus data barang</p>
--	---

Sumber : Hasil Penelitian (2024)

4. Mengelola Barang Gambar

Tabel II. 4
Deskripsi Use Case Mengelola Barang Gambar

Use case name	Mengelola barang gambar
Requirement	User dapat melakukan tambah, edit, dan hapus pada mengelola barang gambar
Goal	User menambah, mengedit, menghapus dan mengakses data barang gambar
Pre-conditions	User telah melakukan login
Post-conditions	User mengakses halaman barang gambar
Failed end condition	User tidak dapat mengakses halaman barang gambar
Primary Actor	Admin
Main Flow / Basic Path	<p>a. User melakukan tambah pada detail data barang gambar</p> <p>1) User mengakses halaman web admin barang gambar</p> <p>2) User mengklik tombol detail</p> <p>3) Sistem menampilkan pada modal form untuk mengisi gambar</p> <p>4) User mengisi form gambar</p> <p>5) User mengklik tombol simpan</p> <p>6) Sistem memverifikasi data</p> <p>7) Sistem menyimpan data barang gambar</p>

	<p>8) Sistem menampilkan pesan “Data barang gambar disimpan”</p> <p>9) Sistem menampilkan data yang diubah berbentuk tabel</p> <p>b. User melakukan edit pada detail data barang gambar</p> <ol style="list-style-type: none"> 1) User mengakses halaman web admin barang 2) User mengklik tombol edit yang dipilih 3) Sistem menampilkan data pada modal form untuk mengubah gambar 4) User mengubah data form gambar 5) User mengklik tombol simpan 6) Sistem memverifikasi data 7) Sistem mengubah data barang gambar 8) Sistem menampilkan pesan “Data barang gambar diubah” 9) Sistem menampilkan data yang diubah berbentuk tabel <p>c. User melakukan hapus data barang gambar</p> <ol style="list-style-type: none"> 1) User mengakses halaman web admin barang 2) User mengklik tombol hapus yang dipilih 3) Sistem menampilkan dialog konfirmasi dengan memiliki pesan “Apakah Kamu Yakin? ingin menghapus data ini!” 4) User mengklik ya
--	--

	<p>5) Sistem memverifikasi data</p> <p>6) Sistem menghapus data barang gambar</p> <p>7) Sistem menampilkan pesan “Data barang gambar dihapus”</p> <p>8) Sistem menghapus data yang ada pada tabel</p>
Altenative Flow	<p>a. Sistem dapat melakukan tambah data pada barang gambar</p> <p>b. Sistem dapat melakukan edit data pada barang gambar</p> <p>c. Sistem dapat melakukan hapus data pada barang gambar. Sistem dapat melakukan hapus data pada barang gambar</p>
Invariant B	<p>a. User melakukan tambah pada detail data barang gambar</p> <p>1) User mengakses halaman web admin barang</p> <p>2) User mengklik tombol detail</p> <p>3) Sistem menampilkan pada modal form untuk mengisi gambar</p> <p>4) User mengosongkan form gambar</p> <p>5) User mengklik tombol simpan</p> <p>6) Sistem memberikan validasi pada form yang wajib diisi</p> <p>7) User mengisi ulang form yang kosong</p> <p>8) User mengklik tombol simpan</p>

	<p>9) Sistem memverifikasi data</p> <p>10) Sistem menyimpan data barang gambar</p> <p>11) Sistem menampilkan pesan “Data barang gambar ditambahkan”</p> <p>12) Sistem menampilkan data yang disimpan berbentuk tabel</p> <p>b. User melakukan edit pada detail data barang gambar</p> <p>1) User mengakses halaman web admin barang gambar</p> <p>2) User mengklik tombol edit yang dipilih</p> <p>3) Sistem menampilkan data pada modal form untuk mengubah gambar</p> <p>4) User mengubah data form gambar</p> <p>5) User mengklik tombol simpan</p> <p>6) Sistem memberikan validasi pada form yang wajib diisi</p> <p>7) User mengisi ulang form yang kosong</p> <p>8) User mengklik tombol simpan</p> <p>9) Sistem memverifikasi data</p> <p>10) Sistem mengubah data barang gambar</p> <p>11) Sistem menampilkan pesan “Data barang gambar diubah”</p> <p>12) Sistem menampilkan data yang diubah berbentuk tabel</p>
--	---

	<p>c. User melakukan hapus data barang gambar</p> <ol style="list-style-type: none"> 1) User mengakses halaman web admin barang gambar 2) User mengklik tombol hapus yang dipilih 3) Sistem menampilkan dialog konfirmasi dengan memiliki pesan “Apakah Kamu Yakin? ingin menghapus data ini!” 4) User mengklik tidak 5) Sistem tidak menghapus data barang gambar
--	---

Sumber : Hasil Penelitian (2024)

5. Mengelola Kategori

Tabel II. 5

Deskripsi Use Case Mengelola Galeri

Use Case Name	Mengelola Galeri
Requirements	User dapat melakukan tambah, edit, dan hapus pada mengelola galeri
Goal	User menambah, mengedit, menghapus dan mengakses data galeri
Pre-conditions	User telah melakukan login
Post-conditions	User mengakses halaman galeri
Failed end condition	User tidak dapat mengakses halaman galeri
Primary Actors	Admin
Main Flow / Basic Path	<p>a. User melakukan tambah data galeri</p> <ol style="list-style-type: none"> 1) User mengakses halaman web admin galeri

	<p>2) User mengklik tombol tambah</p> <p>3) Sistem menampilkan modal form untuk mengisi nama, tanggal mulai, tanggal selesai dan deskripsi</p> <p>4) User mengisi data form mengisi nama, tanggal mulai, tanggal selesai dan deskripsi</p> <p>5) User mengklik tombol simpan</p> <p>6) Sistem memverifikasi data</p> <p>7) Sistem menyimpan data galeri</p> <p>8) Sistem menampilkan pesan “Data galeri ditambahkan”</p> <p>9) Sistem menampilkan data yang disimpan berbentuk tabel</p> <p>b. User melakukan edit data galeri</p> <p>1) User mengakses halaman web admin galeri</p> <p>2) User mengklik tombol edit yang dipilih</p> <p>3) Sistem menampilkan data pada modal form untuk mengubah nama, tanggal mulai, tanggal selesai dan deskripsi</p> <p>4) User mengubah data form nama, tanggal mulai, tanggal selesai dan deskripsi</p> <p>5) User mengklik tombol simpan</p> <p>6) Sistem memverifikasi data</p> <p>7) Sistem mengubah data galeri</p> <p>8) Sistem menampilkan pesan “Data galeri diubah”</p>
--	--

	<p>9) Sistem menampilkan data yang diubah berbentuk tabel</p> <p>c. User melakukan hapus data galeri</p> <ol style="list-style-type: none"> 1) User mengakses halaman web admin galeri 2) User mengklik tombol hapus yang dipilih 3) Sistem menampilkan dialog konfirmasi dengan memiliki pesan “Apakah Kamu Yakin? ingin menghapus data ini!” 4) User mengklik ya 5) Sistem memverifikasi data 6) Sistem menghapus data galeri 7) Sistem menampilkan pesan “Data galeri dihapus” 8) Sistem menghapus data yang ada pada tabel
Alternative Flow	<p>a. Sistem dapat melakukan tambah data pada galeri</p> <p>b. Sistem dapat melakukan edit data pada galeri</p> <p>c. Sistem dapat melakukan hapus data pada galeri</p>
Invariant B	<p>a. User melakukan tambah data galeri</p> <ol style="list-style-type: none"> 1) User mengakses halaman web admin galeri 2) User mengklik tombol tambah 3) Sistem menampilkan modal form untuk mengisi nama, tanggal mulai, tanggal selesai dan deskripsi 4) User mengisi data form mengisi nama, tanggal mulai, tanggal selesai dan deskripsi 5) User mengklik tombol simpan

	<p>6) Sistem memberikan validasi pada form yang wajib diisi</p> <p>7) User mengisi ulang form yang kosong</p> <p>8) User mengklik tombol simpan</p> <p>9) Sistem memverifikasi data</p> <p>10) Sistem menyimpan data galeri</p> <p>11) Sistem menampilkan pesan “Data galeri ditambahkan”</p> <p>12) Sistem menampilkan data yang disimpan berbentuk tabel</p> <p>b. User melakukan edit data galeri</p> <p>1) User mengakses halaman web admin galeri</p> <p>2) User mengklik tombol edit yang dipilih</p> <p>3) Sistem menampilkan data pada modal form untuk mengubah nama, tanggal mulai, tanggal selesai dan deskripsi</p> <p>4) User mengubah data form nama, tanggal mulai, tanggal selesai dan deskripsi</p> <p>5) User mengklik tombol simpan</p> <p>6) Sistem memberikan validasi pada form yang wajib diisi</p> <p>7) User mengisi ulang form yang kosong</p> <p>8) User mengklik tombol simpan</p> <p>9) Sistem memverifikasi data</p>
--	--

	<p>10) Sistem mengubah data galeri</p> <p>11) Sistem menampilkan pesan “Data galeri diubah”</p> <p>12) Sistem menampilkan data yang diubah berbentuk tabel</p> <p>c. User melakukan hapus data galeri</p> <p>1) User mengakses halaman web admin galeri</p> <p>2) User mengklik tombol hapus yang dipilih</p> <p>3) Sistem menampilkan dialog konfirmasi dengan memiliki pesan “Apakah Kamu Yakin? ingin menghapus data ini!”</p> <p>4) User mengklik tidak</p> <p>5) Sistem tidak menghapus data galeri</p>
--	--

Sumber : Hasil Penelitian (2024)

6. Mengelola Galeri Gambar

Tabel II. 6

Deskripsi Use Case Mengelola Galeri Gambar

Use Case Name	Mengelola galeri gambar
Requirements	User dapat melakukan tambah, edit, dan hapus pada mengelola galeri gambar
Goal	User menambah, mengedit, menghapus dan mengakses data galeri gambar
Pre-conditions	User telah melakukan login
Post-conditions	User mengakses halaman detail galeri gambar

Failed end condition	User tidak dapat mengakses halaman detail galeri gambar
Primary Actors	Admin
Main Flow / Basic Path	<p>a. User melakukan tambah data detail galeri gambar</p> <ol style="list-style-type: none"> 1) User mengakses halaman web admin detail 2) User mengklik tombol tambah 3) Sistem menampilkan modal form untuk gambar 4) User mengisi data form gambar 5) User mengklik tombol simpan 6) Sistem memverifikasi data 7) Sistem menyimpan data detail galeri gambar 8) Sistem menampilkan pesan “Data galeri gambar ditambahkan” 9) Sistem menampilkan data yang disimpan berbentuk tabel <p>b. User melakukan edit data detail galeri gambar</p> <ol style="list-style-type: none"> 1) User mengakses halaman web admin detail galeri gambar 2) User mengklik tombol edit yang dipilih 3) Sistem menampilkan data pada modal form untuk mengubah gambar 4) User mengubah data form gambar 5) User mengklik tombol simpan 6) Sistem memverifikasi data

	<p>7) Sistem mengubah data detail galeri gambar</p> <p>8) Sistem menampilkan pesan “Data galeri gambar diubah”</p> <p>9) Sistem menampilkan data yang diubah berbentuk tabel</p> <p>c. User melakukan hapus data detail galeri gambar</p> <p>1) User mengakses halaman web admin detail galeri gambar</p> <p>2) User mengklik tombol hapus yang dipilih</p> <p>3) Sistem menampilkan dialog konfirmasi dengan memiliki pesan “Apakah Kamu Yakin? ingin menghapus data ini?”</p> <p>4) User mengklik ya</p> <p>5) Sistem memverifikasi data</p> <p>6) Sistem menghapus data galeri gambar</p> <p>7) Sistem menampilkan pesan “Data galeri gambar dihapus”</p> <p>8) Sistem menghapus data yang ada pada tabel</p>
Alternative Flow	<p>a. Sistem dapat melakukan tambah data pada detail galeri gambar</p> <p>b. Sistem dapat melakukan edit data pada detail galeri gambar</p> <p>c. Sistem dapat melakukan hapus data pada detail galeri gambar</p>

Invariant B	<p>a. User melakukan tambah data detail galeri gambar</p> <ol style="list-style-type: none"> 1) User mengakses halaman web admin detail galeri gambar 2) User mengklik tombol tambah 3) Sistem menampilkan modal form untuk gambar 4) User mengisi data form mengisi nama, tanggal mulai, tanggal selesai dan deskripsi 5) User mengklik tombol simpan 6) Sistem memberikan validasi pada form yang wajib diisi 7) User mengisi ulang form yang kosong 8) User mengklik tombol simpan 9) Sistem memverifikasi data 10) Sistem menyimpan data galeri gambar 11) Sistem menampilkan pesan “Data galeri gambar ditambahkan” 12) Sistem menampilkan data yang disimpan berbentuk tabel <p>b. User melakukan edit data detail galeri gambar</p> <ol style="list-style-type: none"> 1) User mengakses halaman web admin detail galeri gambar 2) User mengklik tombol edit yang dipilih
-------------	--

	<p>3) Sistem menampilkan data pada modal form untuk mengubah nama, tanggal mulai, tanggal selesai dan deskripsi</p> <p>4) User mengubah data form nama, tanggal mulai, tanggal selesai dan deskripsi</p> <p>5) User mengklik tombol simpan</p> <p>6) Sistem memberikan validasi pada form yang wajib diisi</p> <p>7) User mengisi ulang form yang kosong</p> <p>8) User mengklik tombol simpan</p> <p>9) Sistem memverifikasi data</p> <p>10) Sistem mengubah data galeri gambar</p> <p>11) Sistem menampilkan pesan “Data galeri gambar diubah”</p> <p>12) Sistem menampilkan data yang diubah berbentuk tabel</p> <p>c. User melakukan hapus data detail galeri gambar</p> <p>1) User mengakses halaman web admin detail galeri gambar</p> <p>2) User mengklik tombol hapus yang dipilih</p> <p>3) Sistem menampilkan dialog konfirmasi dengan memiliki pesan “Apakah Kamu Yakin? ingin menghapus data ini!”</p> <p>4) User mengklik tidak</p>
--	--

	5) Sistem tidak menghapus data galeri gambar
--	--

Sumber : Hasil Penelitian (2024)

7. Mengelola Transaksi

Tabel II. 7

Deskripsi Use Case Mengelola Transaksi

Use case name	Mengelola transaksi
Requirement	Telah melakukan login
Goal	Dapat mengedit, menghapus data transaksi
Pre-conditions	login
Post-conditions	Data transaksi berhasil diperbarui, atau dihapus dari basis data
Failed end condition	Data transaksi tidak valid
Primary Actor	Admin
Main Flow / Basic Path	<ol style="list-style-type: none"> 1.Admin mengakses halaman transaksi 2.Admin memilih untuk mengedit atau menghapus data 3.Admin memasukan atau memperbarui data transaksi 4.Sistem memeriksa validasi data masukan 5.Jika data valid sistem menyimpan atau memperbarui data di basis data 6.Sistem menampilkan pesan “Data berhasil diperbarui” atau “Data berhasil disimpan” 7.Jika data dihapus, sistem menampilkan pesan “Data berhasil dihapus”
Altenative Flow	<ol style="list-style-type: none"> 1. Admin memasukan atau memperbarui data transaksi

	<p>2. Sistem memeriksa validasi data masukan</p> <p>3. Jika data tidak valid, sistem menampilkan pesan “Data masukan harus diisi”</p> <p>4. Admin memperbaiki data masukan</p> <p>5. Sistem kembali memeriksa validasi data</p> <p>6. Jika data valid, sistem menyimpan atau memperbarui data di basis data</p> <p>7. Sistem menampilkan pesan “Data berhasil disimpan” atau “Data berhasil diperbarui”</p>
--	---

Sumber : Hasil Penelitian (2024)

8. Mengubah Pengaturan

Tabel II. 8

Deskripsi Use Case Mengubah Pengaturan

Use case name	Mengubah pengaturan
Requirement	User dapat melakukan edit data pengaturan
Goal	User menggunakan fitur pengaturan
Pre-conditions	User telah melakukan login
Post-conditions	User mengakses halaman pengaturan
Failed end condition	User tidak dapat mengakses halaman pengaturan
Primary Actor	Admin
Main Flow / Basic Path	<p>1. Admin mengakses halaman pengaturan</p> <p>2. Admin mengedit data pengaturan</p> <p>3. Admin memasukan atau memperbarui data pengaturan</p>

	<p>4. Sistem memeriksa validasi data masukan</p> <p>5. Jika data valid sistem menyimpan atau memperbarui data di basis data</p> <p>6. Sistem menampilkan pesan “Data berhasil diperbarui” atau “Data berhasil disimpan”</p>
Altenative Flow	Sistem dapat melakukan edit data pada pengaturan
Invariant B	<p>User melakukan edit data pengaturan</p> <p>1) User mengakses halaman web admin pengaturan</p> <p>2) Sistem menampilkan data nama, email, no hp, alamat dan deskripsi.</p> <p>4) User mengubah data form nama, email, no hp, alamat dan deskripsi. dengan salah satu form ada yang kosong</p> <p>5) User mengklik tombol simpan</p> <p>6) Sistem memberikan validasi pada form yang wajib diisi</p> <p>7) User mengisi ulang form yang kosong</p> <p>8) User mengklik tombol simpan</p> <p>9) Sistem memverifikasi data</p> <p>10) Sistem mengubah data barang bawah</p> <p>11) Sistem menampilkan pesan “Data pengaturan diubah”</p>

Sumber : Hasil Penelitian (2024)

9. Mengubah Profil

Tabel II. 9
Deskripsi Use Case Mengelola Profil

Use case name	Mengubah profile
Requirement	User dapat melakukan edit profile dan ubah password
Goal	User menggunakan fitur edit profile dan ubah password
Pre-conditions	User telah melakukan login
Post-conditions	User mengakses halaman Profile
Failed end condition	User tidak dapat mengakses halaman Profile
Primary Actor	Admin, User
Main Flow / Basic Path	<ol style="list-style-type: none"> 1. User mengakses halaman profile 2. User memilih untuk mengedit data profil dan ubah password 3. User memasukan data profil dan memasukan data ubah password 4. Sistem memeriksa validasi data masukan 5. Jika data valid sistem menyimpan atau memperbarui data di basis data 6. Sistem menampilkan pesan “Data berhasil disimpan”
Altenative Flow	<ol style="list-style-type: none"> a. Sistem dapat melakukan edit profile b. Sistem dapat melakukan ubah password
Invariant B	<ol style="list-style-type: none"> 1. User memasukan data profil dan memasukan data ubah password 2. Sistem memeriksa validasi data masukan

	<p>3. Jika data tidak valid, sistem menampilkan pesan “Data masukan harus diisi”</p> <p>4. User memperbaiki data masukan</p> <p>5. Sistem kembali memeriksa validasi data</p> <p>6. Jika data valid, sistem menyimpan atau memperbarui data di basis data</p>
--	---

Sumber : Hasil Penelitian (2024)

10. Melihat Tentang pada Halaman User

Tabel II. 10

Deskripsi Use Case Tentang

Use Case Name	Melihat tentang
Requirements	User dapat mengakses fitur tentang
Goal	User menggunakan fitur tentang
Pre-conditions	User telah melakukan login atau telah membuka situs web
Post-conditions	User mengakses halaman tentang
Failed end condition	User tidak dapat mengakses halaman tentang
Primary Actors	User
Main Flow / Basic Path	<ol style="list-style-type: none"> 1. User membuka menu tentang. 2. Sistem menampilkan halaman tentang
Alternative Flow	-

Sumber : Hasil Penelitian (2024)

11. Melihat Galeri pada halaman User

Tabel II. 11

Deskripsi Use Case Melihat Galeri

Use Case Name	Melihat galeri
Requirements	User dapat mengakses fitur galeri
Goal	User menggunakan fitur galeri
Pre-conditions	User telah melakukan login atau telah membuka situs web
Post-conditions	User mengakses halaman galeri
Failed end condition	User tidak dapat mengakses halaman galeri
Primary Actors	User
Main Flow / Basic Path	<ol style="list-style-type: none">1. User membuka menu galeri.2. Sistem menampilkan daftar galeri yang tersedia.3. User memilih galeri yang ingin dilihat.4. Sistem menampilkan detail galeri yang dipilih oleh user5. User mencari nama galeri yang ingin dicari6. Sistem menampilkan galeri sesuai yang dicari
Alternative Flow	Sistem dapat melakukan pencarian barang

Sumber : Hasil Penelitian (2024)

12. Melihat Barang pada halaman User

Tabel II. 12

Deskripsi Use Case Melihat Barang

Use Case Name	Melihat barang
Requirements	User dapat mengakses fitur barang
Goal	User menggunakan fitur barang
Pre-conditions	User telah melakukan login atau telah membuka situs web
Post-conditions	User mengakses halaman barang
Failed end condition	User tidak dapat mengakses halaman barang
Primary Actors	User
Main Flow / Basic Path	<ol style="list-style-type: none">a. Halaman awal barang<ol style="list-style-type: none">1. User membuka menu barang.2. Sistem menampilkan daftar barang yang tersedia.3. User mencari nama barang yang ingin dicari4. Sistem menampilkan barang sesuai yang dicari5. User memilih barang yang ingin dilihat.6. User menekan tombol detail7. Sistem menampilkan detail barang8. User menekan tombol beli9. Sistem menampilkan halaman login/registrasib. Halaman barang sesuai kategori barang<ol style="list-style-type: none">1. User memilih kategori

	<ol style="list-style-type: none"> 2. Sistem menampilkan barang sesuai kategori 3. User mencari nama barang yang ingin dicari 4. Sistem menampilkan barang sesuai yang dicari 5. User memilih barang yang ingin dilihat. 6. User menekan tombol detail 7. Sistem menampilkan detail barang 8. User menekan tombol beli 9. Sistem menampilkan halaman login/registrasi
Alternative Flow	Sistem dapat melakukan pencarian barang

Sumber : Hasil Penelitian (2024)

13. Melakukan Transaksi Barang

Tabel II. 13

Deskripsi Use Case Melakukan Transaksi Barang

Use Case Name	Melakukan transaksi barang
Requirements	User dapat mengakses fitur transaksi, user dapat melakukan transaksi
Goal	User menggunakan fitur transaksi
Pre-conditions	User telah melakukan login
Post-conditions	User mengakses halaman transaksi
Failed end condition	User tidak dapat mengakses halaman transaksi
Primary Actors	User
Main Flow / Basic Path	<ol style="list-style-type: none"> a. Halaman awal barang 1. User membuka menu barang.

	<ol style="list-style-type: none"> 2. Sistem menampilkan daftar barang yang tersedia. 3. User memilih barang yang ingin dibeli. 4. User menekan tombol beli untuk memasukkan barang ke keranjang 5. User melakukan transaksi pada halaman keranjang 6. User melakukan pembayaran menggunakan <i>payment gateway</i> 7. User melihat status transaksi pada halaman transaksi <p>b. Halaman barang sesuai kategori barang</p> <ol style="list-style-type: none"> 1. User melihat barang sesuai dengan kategori yang ingin dilihat 2. Sistem menampilkan barang sesuai kategori 3. User memilih barang yang ingin dibeli. 4. User menekan tombol beli untuk memasukkan barang ke keranjang 5. User melakukan transaksi pada halaman keranjang 6. User melakukan pembayaran menggunakan <i>payment gateway</i> 7. User melihat status transaksi pada halaman transaksi
--	---

	<p>c. Halaman Pencarian Barang</p> <ol style="list-style-type: none"> 1. User membuka menu barang 2. Sistem menampilkan daftar barang yang tersedia 3. User mencari nama barang yang ingin dicari 4. Sistem menampilkan barang sesuai yang dicari 5. User memilih barang yang ingin dilihat. 8. User menekan tombol beli untuk memasukkan barang ke keranjang 9. User melakukan transaksi pada halaman keranjang 10. User melakukan pembayaran menggunakan <i>payment gateway</i> 11. User melihat status transaksi pada halaman transaksi
Alternative Flow	Sistem melakukan pencarian barang
Invariant B	<p>a. Halaman awal barang</p> <ol style="list-style-type: none"> 1. User membuka menu barang. 2. Sistem menampilkan daftar barang yang tersedia. 3. User memilih barang yang ingin dibeli. 4. User hanya memasukkan barang ke keranjang 5. User tidak melakukan transaksi <p>b. Halaman barang sesuai kategori barang</p>

	<ol style="list-style-type: none"> 1. User melihat barang sesuai dengan kategori yang ingin dilihat 2. Sistem menampilkan barang sesuai kategori 3. User memilih barang yang ingin dibeli. 4. User hanya memasukkan barang ke keranjang 5. User tidak melakukan transaksi <p>c. Halaman Pencarian Barang</p> <ol style="list-style-type: none"> 1. User membuka menu barang 2. Sistem menampilkan daftar barang yang tersedia 3. User mencari nama barang yang ingin dicari 4. Sistem menampilkan barang sesuai yang dicari 5. User memilih barang yang ingin dibeli. 6. User hanya memasukkan barang ke keranjang 7. User tidak melakukan transaksi
--	--

Sumber : Hasil Penelitian (2024)

14. Melihat Halaman Keranjang

Tabel II. 14

Deskripsi Use Case Melihat Keranjang

Use Case Name	Melihat halaman keranjang
Requirements	User dapat mengakses fitur keranjang
Goal	User menggunakan fitur keranjang
Pre-conditions	User telah melakukan login
Post-conditions	User mengakses halaman keranjang
Failed end condition	User tidak dapat mengakses halaman keranjang
Primary Actors	User
Main Flow / Basic Path	<ol style="list-style-type: none">1. User membuka menu keranjang2. Sistem menampilkan halaman keranjang user3. Sistem menampilkan daftar produk yang akan dibeli user4. User melakukan edit quantity pada aksi edit5. User melakukan pembayaran dengan <i>payment midtrans</i>
Alternative Flow	-
Invariant B	<ol style="list-style-type: none">1. User membuka menu keranjang2. Sistem menampilkan halaman keranjang user3. Sistem menampilkan daftar produk yang akan dibeli4. User melakukan hapus per barang pada halaman keranjang5. User tidak melakukan pembayaran

Sumber : Hasil Penelitian (2024)

15. Melihat Halaman Transaksi

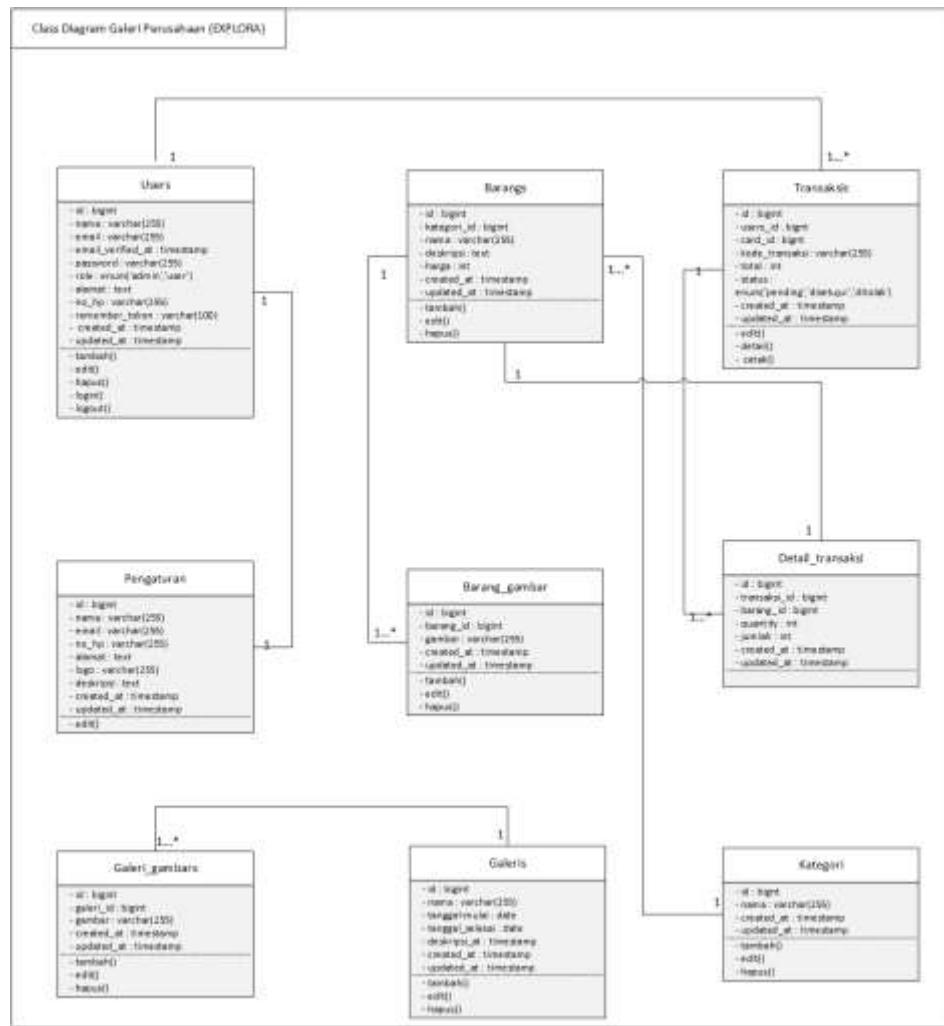
Tabel II. 15
Deskripsi Use Case Melihat Transaksi

Use Case Name	Melihat halaman transaksi
Requirements	User dapat mengakses fitur transaksi
Goal	User menggunakan fitur transaksi
Pre-conditions	User telah melakukan login
Post-conditions	User mengakses halaman transaksi
Failed end condition	User tidak dapat mengakses halaman transaksi
Primary Actors	User
Main Flow / Basic Path	<ol style="list-style-type: none">1. User membuka menu transaksi2. Sistem menampilkan halaman transaksi user3. Sistem menampilkan perubahan pada status transaksi4. User membuka detail transaksi pada kolom aksi5. Sistem menampilkan halaman detail data transaksi6. User mencetak laporan pertransaksi7. User mencetak laporan transaksi per periode
Alternative Flow	-

Sumber : Hasil Penelitian (2024)

B. Rancangan Class Diagram

Menurut (Anjani Fatmasari, 2024) Spesifikasi "kelas" adalah spesifikasi yang, saat diinstansiasi, menghasilkan sebuah objek. Ini adalah dasar dalam pengembangan dan desain berbasis objek. Klasifikasi mendefinisikan atribut sistem dan memberikan layanan untuk mengelola atribut tersebut melalui metode.



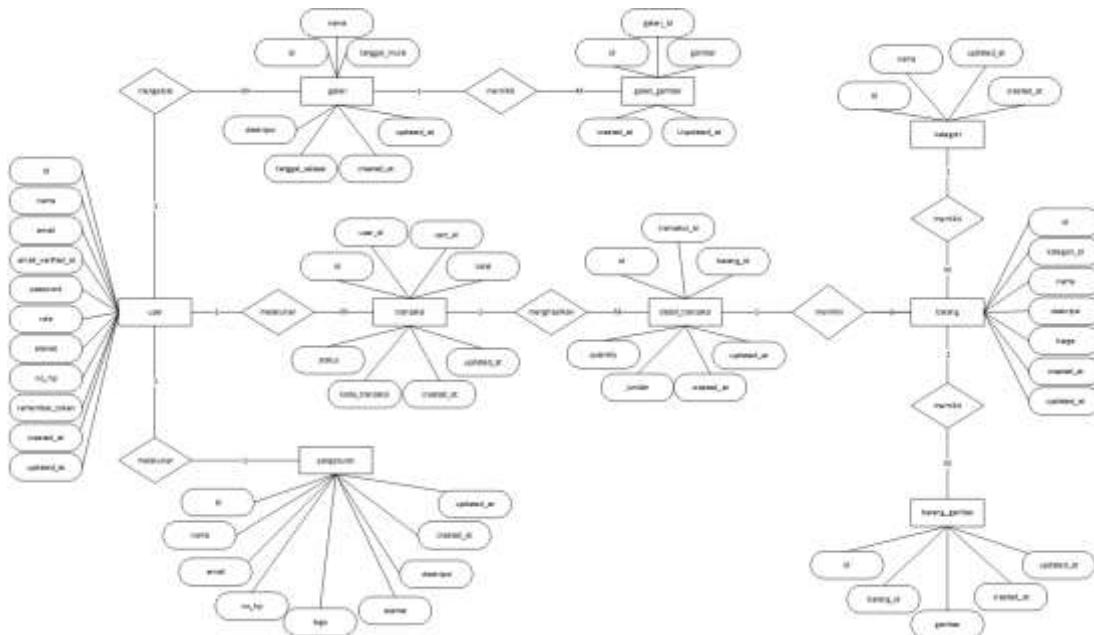
Gambar II. 2 Class Diagram Esa Gallery

Sumber : Hasil Penelitian (2024)

2.2 Rancangan Database

A. Entity Relationship Diagram (ERD)

Menurut (Batara et al., 2024) Model hubungan entitas yang terdiri dari komponen kumpulan hubungan dikenal sebagai Entity Relationship Diagram. Setiap bagian diberi sifat yang mewakili seluruh keadaan dunia nyata dan digambarkan secara lebih sistematis melalui diagram.



Gambar II. 3 Entity Relationship Diagram Esa Gallery

Sumber : Hasil Penelitian (2024)

Gambar II.3 merupakan Entity Relationship Diagram (ERD) untuk Sistem Informasi Esa Gallery yang merinci hubungan antara entitas utama dalam sistem ini. Diagram ini mencangkup sembilan entitas: user, transaksi, detail transaksi, barang, barang gambar, kategori, galeri, galeri gambar dan pengaturan. Relasi antar entitas digambarkan dengan jelas dalam ERD ini. Dengan menggabungkan entitas dan relasi ini, ERD memberikan visualisasi yang jelas dan terstruktur mengenai bagaimana data diorganisir dan dikelola dalam Sistem Informasi Esa Gallery, serta menunjukkan

bagaimana entitas-entitas ini berinteraksi satu sama lain untuk mendukung operasi galeri perusahaan secara keseluruhan.

B. Spesifikasi File

Berdasarkan rancangan ERD pada poin A selanjutnya dilakukan spesifikasi file yang mana file – file ini ada pada database dengan nama esagallery dengan parameter – parameter sebagai berikut :

1. Spesifikasi File User

Nama File	:	User
Akronim	:	Users
Fungsi	:	Untuk menyimpan data user
Tipe File	:	File Master
Organisasi File	:	Indexed sequential
Akses File	:	Random
Media	:	SSD
Panjang record	:	140
Kunci Field	:	id
Software	:	mysql

No	Elemen Data	Nama Field	Tipe	Size	Ket
1	Id	id	bigint	20	Primary Key
2	Nama	nama	Varchar	30	
3	Email	email	Varchar	30	Unique
4	Email	email_verified_at	timestamp		
5	Password	password	Varchar	30	

6	Remember Token	Remember_token	Varchar	30	
7	Created at	creadted_at	timestamp		
8	Updated at	updated_at	timestamp		

Tabel II. 16

Spesifikasi File User

Sumber : Hasil Penelitian (2024)

2. Spesifikasi File Barang



Nama File	: Barang
Akronim	: Barang
Fungsi	: Untuk menyimpan data barang
Tipe File	: File Master
Organisasi File	: Indexed sequential
Akses File	: Random
Media	: SSD
Panjang record	: 81
Kunci Field	: id
Software	: mysql

No	Elemen Data	Nama Field	Tipe	Size	Ket
1	Id	id	bigint	20	Primary key
2	Kategori Id	kategori_id	bigint	20	Foreign key
3	Nama	nama	varchar	30	
4	Deskripsi	deskripsi	Text		
5	Harga	harga	Int	11	

6	Created at	created_at	timestampt		
7	Updated at	updated_at	timestampt		

Tabel II. 17

Spesifikasi File Barang

Sumber : Hasil Penelitian (2024)

3. Spesifikasi File Barang_Gambar

Nama File	: Barang_Gambar
Akronim	: Barang_gambar
Fungsi	: Untuk menyimpan data barang gambar
Tipe File	: File Master
Organisasi File	: Indexed sequential
Akses File	: Random
Media	: SSD
Panjang record	: 70
Kunci Field	: id
Software	: mysql

No	Elemen Data	Nama Field	Tipe	Size	Ket
1.	Id	id	bigint	20	Primary key
2.	Barang Id	barang_id	bigint	20	Foreign key
3.	Gambar	gambar	Varchar	30	
4.	Created at	created_at	timestampt		
5.	Updated at	updated_at	timestampt		

Tabel II. 18

Spesifikasi File Barang_gambar

Sumber : Hasil Penelitian (2024)

4. Spesifikasi File Galeri

Nama File : Galeri
 Akronim : Galeri
 Fungsi : Untuk menyimpan data galeri
 Tipe File : File Master
 Organisasi File : Indexed sequential
 Akses File : Random
 Media : SSD
 Panjang record : 70
 Kunci Field : id
 Software : mysql

No	Elemen Data	Nama Field	Tipe	Size	Ket
1.	Id	id	bigint	20	Primary key
2.	Kategori Id	kategori_id	bigint	20	Foreign key
3.	Nama	nama	varchar	30	
4.	Tanggal Mulai	tanggal_mulai	date		
5.	Tanggal Selesai	tanggal_selesai	date		
6.	Deskripsi	deskripsi	text		
7.	Created at	created_at	timestamp		
8.	Updated at	updated_at	timestamp		

Tabel II. 19

Spesifikasi File Galeri

Sumber : Hasil Penelitian (2024)

5. Spesifikasi File Galeri_Gambar

Nama File : Galeri Gambar
Akrоним : Galeri_gambar
Fungsi : Untuk menyimpan data galeri gambar
Tipe File : File Master
Organisasi File : Indexed sequential
Akses File : Random
Media : SSD
Panjang record : 50
Kunci Field : id
Software : mysql

No	Elemen Data	Nama Field	Tipe	Size	Ket
1.	Id	id	bigint	20	Primary key
2.	Galeri Id	galeri_id	Varchar	30	Foreign key
3.	Created at	created_at	timestamp		
4.	Updated at	updated_at	timestamp		

Tabel II. 20

Spesifikasi File Galeri_Gambar

Sumber : Hasil Penelitian (2024)

6. Spesifikasi File Kategori

Nama File : Kategori
Akrоним : Kategori
Fungsi : Untuk menyimpan data kategori
Tipe File : File Master

Organisasi File : Indexed sequential
 Akses File : Random
 Media : SSD
 Panjang record : 50
 Kunci Field : id
 Software : mysql

No	Elemen Data	Nama Field	Tipe	Size	Ket
1.	Id	Id	bigint	20	Primary key
2.	Nama	Nama	varchar	30	
3.	Created at	created_at	timestamp		
4.	Updated at	updated_at	timestamp		

Tabel II. 21

Spesifikasi File Kategori

Sumber : Hasil Penelitian (2024)

7. Spesifikasi File Transaksi

Nama File : Transaksi
 Akronim : Transaksi
 Fungsi : Untuk menyimpan data transaksi
 Tipe File : File Master
 Organisasi File : Indexed sequential
 Akses File : Random
 Media : SSD
 Panjang record : 191
 Kunci Field : id

Software : mysql

No	Elemen Data	Nama Field	Tipe	Size	Ket
1	Id	id	bigint	20	Primary key
2	Barang Id	barang_id	bigint	20	Foreign key
3	User Id	user_id	bigint	20	Foreign key
4	Cart Id	cart_id	bigint	20	Foreign key
5	Nama	nama	varchar	30	
6	Email	email	varchar	30	
7	No Hp	no_hp	varchar	30	
8	Alamat	alamat	text		
9	Total	total	int	11	
10	Status	status	varchar	30	
11	Created at	created_at	timestamp		
12	Updated at	updated_at	timestamp		

Table II.7

Tabel II. 22
Spesifikasi File Transaksi

Sumber : Hasil Penelitian (2024)

8. Spesifikasi File Pengaturan

- | | |
|-----------------|-----------------------------------|
| Nama File | : Pengaturan |
| Akronim | : Pengaturan |
| Fungsi | : Untuk menyimpan data pengaturan |
| Tipe File | : File Master |
| Organisasi File | : Indexed sequential |

Akses File : Random

Media : SSD

Panjang record : 140

Kunci Field : id

Software : mysql

No	Elemen Data	Nama Field	Tipe	Size	Ket
1.	Id	id	bigint	20	Primary Key
2.	Nama	nama	varchar	30	
3.	Email	email	varchar	30	
4.	No Hp	no_hp	varchar	30	
5.	Alamat	alamat	text		
6.	Logo	logo	varchar	30	
7.	Deskripsi	deskripsi	text		
8.	Created at	created_at	timestamp		
9.	Updated at	updated_at	timestamp		

Tabel II. 23

Spesifikasi File Pengaturan

Sumber : Hasil Penelitian (2024)

C. Database Esa Gallery

Untuk menghubungkan website yang menggunakan framework Laravel pada database, perlu dilakukan beberapa konfigurasi melalui file .env. Berikut adalah konfigurasi yang umumnya diperlukan :

DB_CONNECTION=mysql

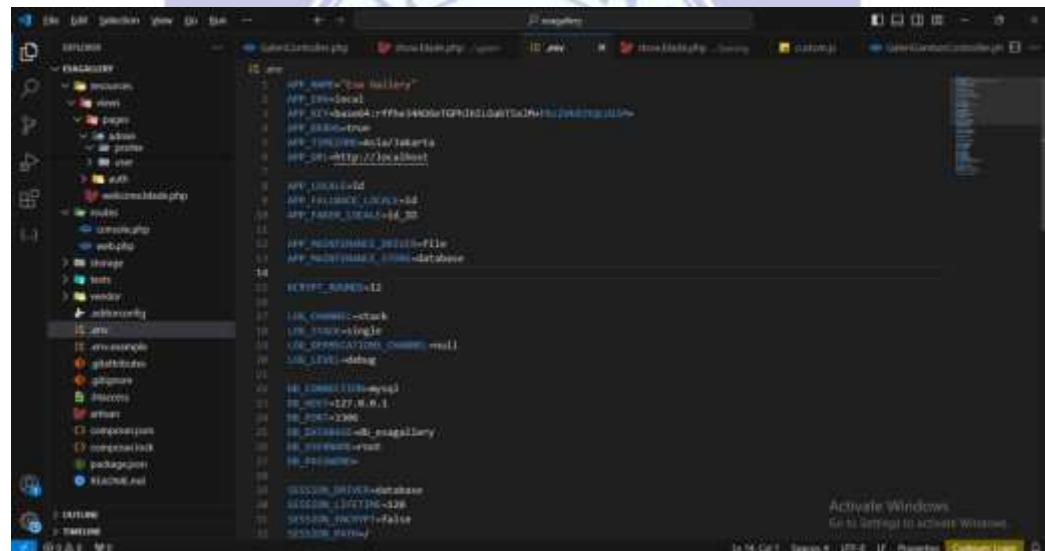
DB HOST=127.0.0.1

DB PORT=3306

DB_DATABASE=db_esagallery

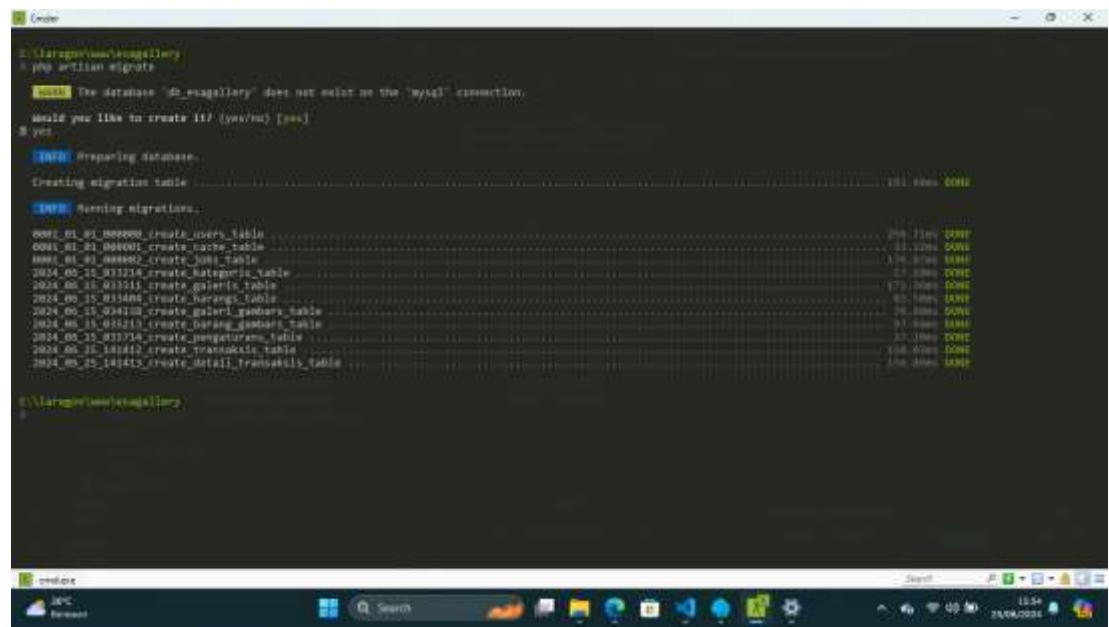
DB USERNAME=root

DB_PASSWORD=



Gambar II. 4 Database Esa Gallery

Membuat database pada proyek Laravel menggunakan terminal dengan perintah “`php artisan migrate`” yang mana kalau database nya belum maka dapat membuat database secara otomatis.



```
c:\laragon\www\magallery> php artisan migrate
[Output]
  The database "do_magallery" does not exist on the "mysql" connection.
Would you like to create it? (yes/no) [yes]
y yes
  Preparing database...
Creating migration table...
Running migrations...
  2024_05_25_085800 create_users_table
  2024_05_25_085801 create_auth_tokens_table
  2024_05_25_085802 create_categories_table
  2024_05_25_085814 create_galleries_table
  2024_05_25_085813 create_gallery_gambar_table
  2024_05_25_085810 create_galeri_gambar_table
  2024_05_25_085811 create_navig_gambar_table
  2024_05_25_085814 create_navig_username_table
  2024_05_25_140442 create_transaksi_table
  2024_05_25_140443 create_detail_transaksi_table
[Output]
Migration completed successfully.
```

Gambar II. 5 konfigurasi rancangan database

Konfigurasi rancangan database pada spesifikasi file dengan membuat migration dari masing – masing tabel diantaranya sebagai berikut :

1. Tabel User

Membuat tabel user yang sudah ada di Laravel sudah ada dengan penerapan sesuai.

```
Schema::create('users', function (Blueprint $table) {
    $table->id();
    $table->string('nama');
    $table->string('email')->unique();
    $table->timestamp('email_verified_at')->nullable();
    $table->string('password');
    $table->enum('role', ['admin', 'user'])->default('user');
    $table->text('alamat')->nullable();
    $table->string('no_hp')->nullable();
    $table->rememberToken();
});
```

```
$table->timestamps();  
});
```

2. Tabel Barang

Membuat tabel barang menggunakan terminal dengan perintah “php artisan make:migration create_barangs_table” yang berelasi dengan tabel barang gambar.

```
Schema::create('barang', function (Blueprint $table) {
```

```
    $table->id();  
  
    $table->foreignId('kategori_id')->constrained();  
  
    $table->string('nama');  
  
    $table->text('deskripsi');  
  
    $table->integer('harga');  
  
    $table->timestamps();  
});
```

3. Tabel Barang_gambar

Membuat tabel barang menggunakan terminal dengan perintah “php artisan make:migration create_gambar_barangs_table”.

```
Schema::create('barang_gambar', function (Blueprint $table) {
```

```
    $table->id();  
  
    $table->foreignId('barang_id')->constrained();  
  
    $table->string('gambar');  
  
    $table->timestamps();  
});
```

4. Tabel Galeri

Membuat tabel barang menggunakan terminal dengan perintah “php artisan make:migration create_galeris_table” yang berelasi dengan tabel galeri barang.

```
Schema::create('galeris', function (Blueprint $table) {  
    $table->id();  
    $table->string('nama');  
    $table->date('tanggal_mulai');  
    $table->date('tanggal_selesai');  
    $table->text('deskripsi');  
    $table->timestamps();  
});
```

5. Tabel Galeri_gambar

Membuat tabel barang menggunakan terminal dengan perintah “php artisan make:migration create_galeri_gambars_table”.

```
Schema::create('galeri_gambars', function (Blueprint $table) {  
    $table->id();  
    $table->foreignId('galeri_id')->constrained();  
    $table->string('gambar');  
    $table->timestamps();  
});
```

6. Tabel Kategori

Membuat tabel barang menggunakan terminal dengan perintah “php artisan make:migration create_kategoris_table” yang berelasi dengan tabel galeri barang.

```
Schema::create('kategoris', function (Blueprint $table) {
```

```
    $table->id();
```

```
    $table->string('nama');
```

```
    $table->timestamps();
```

```
});
```

7. Tabel Transaksi

Membuat tabel barang menggunakan terminal dengan perintah “php artisan make:migration create_transaksis_table”

```
Schema::create('transaksis', function (Blueprint $table) {
```

```
    $table->id();
```

```
    $table->foreignId('users_id')->constrained();
```

```
    $table->string('kode_transaksi')->unique();
```

```
    $table->integer('total');
```

```
    $table->enum('status', ['pending', 'disetujui', 'ditolak']);
```

```
    $table->timestamps();
```

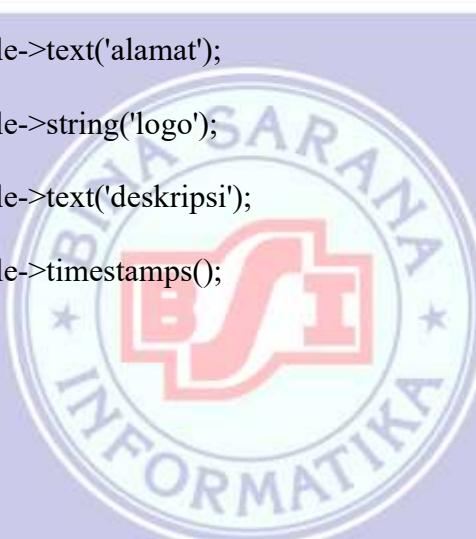
```
});
```

8. Tabel Pengaturan

Membuat tabel barang menggunakan terminal dengan perintah “php artisan make:migration create_pengaturans_table”

```
Schema::create('pengaturans', function (Blueprint $table) {
```

```
    $table->id();  
  
    $table->string('nama');  
  
    $table->string('email');  
  
    $table->string('no_hp');  
  
    $table->text('alamat');  
  
    $table->string('logo');  
  
    $table->text('deskripsi');  
  
    $table->timestamps();  
  
});
```



UNIVERSITAS

2.3 Rancangan Form Master

A. Membuat Form Master User

Tambah Data User

Nama *

Email *

Password *

Konfirmasi Password *

Role *

No Hp *

Alamat *

Batal Simpan

Tambah Data User

Nama *

Email *

Password *

Konfirmasi Password *

Role

Pilih Role

No Hp *

Alamat *

Batal Simpan

Gambar II. 6

Desain Form Master User

Gambar II. 7

Running Form Master User

1. Klik tombol tambah data user
2. Isi Form Master tambah data user

Tambah Data User

Tambah Data User

1
2
3
4
5
6
7
8
9

Gambar II. 8

Gambar II. 9

Form Tambah Data User

Form Master tambah_user

Tabel II. 24

Spesifikasi properties komponen Form Master_Tambah_User

No	Komponen	Properti
1	Form input text 1	nama Teks : (kosongkan)
2	Form input text 2	email Teks : (kosongkan)
3	Form input password 1	password

		Teks : (kosongkan)
4	Form input password 2	Password_confirm Teks : (kosongkan)
5	Form select	role input : (kosongkan)
6	Form input text 3	no_hp Teks : (kosongkan)
7	Form input text 4	Alamat Teks : (kosongkan)
8	Button 1	tombol_simpan Teks : Simpan
9	Button 2	Tombol_batal Teks : Batal

Sumber : Hasil Penelitian (2024)

Logika Jalannya Program

- Masuk ke halaman admin user.
- Klik tombol “Tambah” untuk menampilkan modal form data user.
- Isi data user: nama, email, password, konfirmasi password, no hp dan alamat.
- Jika ada data yang tidak sesuai, sistem memberikan validasi. Jika semua data valid, form disimpan ke database.

3. Script pada tampilan halaman admin user yang mana disini dapat melakukan data user, tambah, edit, dan hapus.

```
@extends('layouts.admin')

@section('title', 'User')

@push('style')
<link rel="stylesheet" href="{{ asset('admin/extensions/datatables.net-bs5/css/dataTables.bootstrap5.min.css') }}">
<link rel="stylesheet" href="{{ asset('admin/compiled/css/table-datatable-jquery.css') }}">
<link rel="stylesheet" href="{{ asset('admin/extensions/sweetalert2/sweetalert2.min.css') }}">
@endpush

@section('main')
<div class="content-wrapper container">
<div class="page-heading">
<div class="row">
<div class="col-12 col-md-6 order-md-1 order-last">
<h3>@yield('title')</h3>
</div>
<div class="col-12 col-md-6 order-md-2 order-first">
```

```

<nav aria-label="breadcrumb" class="breadcrumb-header float-
start float-lg-end">

    <ol class="breadcrumb">

        <li class="breadcrumb-item"><a
            href="/admin">Dashboard</a></li>

        <li class="breadcrumb-item active" aria-
            current="page">@yield('title')</li>

    </ol>

```

</nav>

</div>

</div>

</div>

</div>

<div class="page-content">

<section class="row">

<div class="col-12">

<div class="card">

<div class="card-body">

<div class="d-flex justify-content-between align-items-
center">

<h5 class="card-title mb-0">Data @yield('title')</h5>

<div>

<button class="btn btn-success btn-sm"
 onclick="getModal('createModal')">

<i class="bi bi-plus me-2"></i>Tambah

```

        </button>

        </div>

        </div>

        <hr>

        <div class="table-responsive">

            <table class="table table-bordered table-striped" id="user-
table" width="100%">

                <thead>

                    <tr>

                        <th width="5%">#</th>
                        <th>Nama</th>
                        <th>Email</th>
                        <th>Role</th>
                        <th>No HP</th>
                        <th>Alamat</th>
                        <th width="15%">Aksi</th>
                    </tr>

                </thead>

                <tbody>

                    </tbody>

                </table>

            </div>

            </div>

            </div>

```

```

        </section>

        </div>

        </div>

        @include('pages.admin.user.modal')

@endsection

@push('scripts')
<script src="{!!
asset('admin/extensions/datatables.net/js/jquery.dataTables.min.js') !!}"></script>
<script src="{!! asset('admin/extensions/datatables.net-
bs5/js/dataTables.bootstrap5.min.js') !!}"></script>
<script src="{!! asset('admin/extensions/sweetalert2/sweetalert2.min.js') !!}"></script>
<script>
$(document).ready(function() {
    datatableCall('user-table', '{ route('admin.user.index') }', [
        {
            data: 'DT_RowIndex',
            name: 'DT_RowIndex'
        },
        {
            data: 'nama',
            name: 'nama'
        },
        {
            data: 'email',

```

```

        name: 'email'

    },
    {
        data: 'role',
        name: 'role'

    },
    {
        data: 'no_hp',
        name: 'no_hp'

    },
    {
        data: 'alamat',
        name: 'alamat'

    },
    {
        data: 'aksi',
        name: 'aksi'

    },
});

});
```



```

$("#saveData").submit(function(e) {

    setButtonLoadingState("#saveData .btn.btn-primary", true);

    e.preventDefault();

    const kode = $("#saveData #id").val();

    let url = "{{ route('admin.user.store') }}";
```

```
const data = new FormData(this);

if (kode !== "") {
    data.append("_method", "PUT");
    url = `/admin/user/${kode}`;
}

const successCallback = function(response) {
    setButtonLoadingState("#saveData .btn.btn-primary", false);
    handleSuccess(response, "user-table", "createModal");
};

const errorCallback = function(error) {
    setButtonLoadingState("#saveData .btn.btn-primary", false);
    handleValidationErrors(error, "saveData", ["nama", "email",
    "password", "no_hp", "alamat", "role"]);
};

ajaxCall(url, "POST", data, successCallback, errorCallback);
});

});

</script>

@endpush
```

4. Script pada tampilan modal yang mana disini berfungsi sebagai form tambah dan edit pada user.

```
<div class="modal fade" id="createModal" tabindex="-1" aria-
labelledby="exampleModalLabel" aria-hidden="true"
data-bs-backdrop="static" data-bs-keyboard="false">
<div class="modal-dialog modal-dialog-centered">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title" id="exampleModalLabel"><span id="label-
modal"></span> Data @yield('title')
</h5>
<button type="button" class="btn-close" data-bs-dismiss="modal"
aria-label="Close"></button>
</div>
<form id="saveData" autocomplete="off">
<div class="modal-body">
<input type="hidden" id="id">
<div class="form-group mb-3">
<label for="nama" class="form-label">Nama <span class="text-
danger">*</span></label>
<input type="text" class="form-control" id="nama"
name="nama">
<small class="invalid-feedback" id="errornama"></small>
</div>
<div class="form-group mb-3">
```

```
<label for="email" class="form-label">Email <span class="text-danger">*</span></label>

<input type="email" class="form-control" id="email"
name="email">

<small class="invalid-feedback" id="erroremail"></small>

</div>

<div class="form-group mb-3">

<label for="password" class="form-label">Password <span
class="text-danger">*</span></label>

<input type="password" class="form-control" id="password"
name="password">

<small class="invalid-feedback" id="errorpassword"></small>

</div>

<div class="form-group mb-3">

<label for="password_confirmation" class="form-
label">Konfirmasi Password <span
class="text-danger">*</span></label>

<input type="password" class="form-control"
id="password_confirmation"

name="password_confirmation">

<small class="invalid-feedback"
id="errorpassword_confirmation"></small>

</div>

<div class="form-group mb-3">
```

```

<label for="role" class="form-label">Role <span class="text-
danger"></label>

<select class="form-control" id="role" name="role">

    <option value="">Pilih Role</option>

    <option value="admin">Admin</option>

    <option value="user">User</option>

</select>

</div>

<div class="form-group mb-3">

    <label for="no_hp" class="form-label">No HP <span
class="text-danger">*</span></label>

    <input type="text" class="form-control" id="no_hp"
name="no_hp">

    <small class="invalid-feedback" id="errorno_hp"></small>

</div>

<div class="form-group mb-3">

    <label for="alamat" class="form-label">Alamat <span
class="text-danger">*</span></label>

    <textarea type="text" class="form-control" id="alamat" rows="4"
name="alamat"></textarea>

    <small class="invalid-feedback" id="erroralamat"></small>

</div>

</div>

```

```

<button type="button" class="btn btn-secondary" data-bs-
dismiss="modal">Batal</button>

<button type="submit" class="btn btn-primary">Simpan</button>

</div>

</form>

</div>

</div>

```

5. Script pada proses tambah data user disini ada beberapa validasi untuk data yang diinput untuk dapat mendapatkan data yang valid dan disimpan pada database.



```

public function store(Request $request)
{
    $validator = Validator::make($request->all(), [
        'nama' => 'required|string',
        'email' => 'required|string|email|unique:users,email',
        'password' => 'required|string|min:6|confirmed',
        'no_hp' => 'required',
        'alamat' => 'required',
        'role' => 'required',
    ]);

    if ($validator->fails()) {

```

```
        return $this->errorResponse($validator->errors(), 'Data tidak valid.',  
        422);  
    }  
  
}
```

```
$user = User::create($request->only('nama', 'email', 'password', 'no_hp',  
'alamat', 'role'));
```

```
return $this->successResponse($user, 'Data User ditambahkan.');
```

```
}
```



B. Membuat Form Master Kategori

Gambar II. 10

Desain Form Master Kategori

Gambar II. 11

Running Form Master Kategori

1. Klik tombol tambah untuk tambah data Kategori
2. Isi Form Master tambah data Kategori

Gambar II. 12

Tambah Data Kategori

Gambar II. 13

Form Master_tambah_kategori

Tabel II. 25

Spesifikasi properties komponen Form Master_Tambah_Kategori

No	Komponen	Properti
1	Form input text 1	nama Teks : (kosongkan)
2	Button 1	tombol_simpan

		Teks : Simpan
3	Button 2	tombol_batal Teks : Batal

Sumber : Hasil Penelitian (2024)

Logika Jalannya Program

- a. Masuk ke halaman admin kategori.
 - b. Klik tombol “Tambah” untuk menampilkan modal form data kategori.
 - c. Isi data kategori : nama
 - d. Jika ada data yang tidak sesuai, sistem memberikan validasi. Jika semua data valid, form disimpan ke database
3. Script pada tampilan halaman admin kategori yang mana disini dapat melakukan data kategori, tambah, edit dan hapus.

```
@extends('layouts.admin')

@section('title', 'Kategori')

@push('style')
<link rel="stylesheet" href="{{ asset('admin/extensions/datatables.net-bs5/css/dataTables.bootstrap5.min.css') }}>
<link rel="stylesheet" href="{{ asset('admin/compiled/css/table-datatable-jquery.css') }}>
<link rel="stylesheet" href="{{ asset('admin/extensions/sweetalert2/sweetalert2.min.css') }}>
```

```

@endpush

@section('main')

<div class="content-wrapper container">

    <div class="page-heading">

        <div class="page-title">

            <div class="row">

                <div class="col-12 col-md-6 order-md-1 order-last">

                    <h3>@yield('title')</h3>

                </div>

                <div class="col-12 col-md-6 order-md-2 order-first">

                    <nav aria-label="breadcrumb" class="breadcrumb-header float-
start float-lg-end">

                        <ol class="breadcrumb">

                            <li class="breadcrumb-item"><a
href="/admin">Dashboard</a></li>

                            <li class="breadcrumb-item active" aria-
current="page">@yield('title')</li>

                        </ol>

                    </nav>

                </div>

            </div>

        </div>

        <div class="page-content">

```

```

<section class="row">
    <div class="col-12">
        <div class="card">
            <div class="card-body">
                <div class="d-flex justify-content-between align-items-center">
                    <h5 class="card-title mb-0">Data @yield('title')</h5>
                    <div>
                        <button class="btn btn-success btn-sm"
                               onclick="getModal('createModal')">
                            <i class="bi bi-plus me-2"></i>Tambah
                        </button>
                    </div>
                </div>
                <hr>
                <div class="table-responsive">
                    <table class="table table-bordered table-striped"
                           id="kategori-table" width="100%">
                        <thead>
                            <tr>
                                <th width="5%">#</th>
                                <th>Nama</th>
                                <th width="15%">Aksi</th>
                            </tr>
                        </thead>

```

```
</tbody>

</table>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

@include('pages.admin.kategori.modal')

@endsection

@push('scripts')

<script src="{{ asset('admin/extensions/datatables.net/js/jquery.dataTables.min.js') }}"></script>

<script src="{{ asset('admin/extensions/datatables.net-bs5/js/dataTables.bootstrap5.min.js') }}"></script>

<script src="{{ asset('admin/extensions/sweetalert2/sweetalert2.min.js') }}"></script>

<script>

$(document).ready(function() {

    datatableCall('kategori-table', '{{ route('admin.kategori.index') }}', [

        data: 'DT_RowIndex',

        name: 'DT_RowIndex'

    ]);

});</script>
```

```
        },
        {
            data: 'nama',
            name: 'nama'
        },
        {
            data: 'aksi',
            name: 'aksi'
        },
        ],
        $("#saveData").submit(function(e) {
            setButtonLoadingState("#saveData .btn.btn-primary", true);
            e.preventDefault();
            const kode = $("#saveData #id").val();
            let url = "{{ route('admin.kategori.store') }}";
            const data = new FormData(this);
            if (kode !== "") {
                data.append("_method", "PUT");
                url = `/admin/kategori/${kode}`;
            }
            const successCallback = function(response) {
                setButtonLoadingState("#saveData .btn.btn-primary", false);
            }
        })
    )
}
```

```

        handleSuccess(response, "kategori-table", "createModal");

    };

const errorCallback = function(error) {
    setButtonLoadingState("#saveData .btn.btn-primary", false);
    handleValidationErrors(error, "saveData", ["nama"]);
};

ajaxCall(url, "POST", data, successCallback, errorCallback);

});

});

</script>

@endpush

```

4. Script pada tampilan modal yang mana disini berfungsi sebagai form tambah dan edit pada kategori.

```

<div class="modal fade" id="createModal" tabindex="-1" aria-
labelledby="exampleModalLabel" aria-hidden="true"
data-bs-backdrop="static" data-bs-keyboard="false">
<div class="modal-dialog modal-dialog-centered">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title" id="exampleModalLabel"><span id="label-
modal"></span> Data @yield('title')
</h5>

```

```
<button type="button" class="btn-close" data-bs-dismiss="modal"
aria-label="Close"></button>

</div>

<form id="saveData" autocomplete="off">

<div class="modal-body">

<input type="hidden" id="id">

<div class="form-group mb-3">

<label for="nama" class="form-label">>Nama <span class="text-
danger">*</span></label>

<input type="text" class="form-control" id="nama"
name="nama">

<small class="invalid-feedback" id="errornama"></small>

</div>
</div>

<div class="modal-footer">

<button type="button" class="btn btn-secondary" data-bs-
dismiss="modal">Batal</button>

<button type="submit" class="btn btn-primary">Simpan</button>

</div>

</form>

</div>

</div>
```

5. Script pada proses tambah data kategori disini ada beberapa validasi untuk data yang diinput untuk dapat mendapatkan data yang valid dan disimpan pada database.

```
public function store(Request $request)  
{  
    $validator = Validator::make($request->all(), [  
        'nama' => 'required|string|unique:kategori,nama',  
    ]);  
  
    if ($validator->fails()) {  
        return $this->errorResponse($validator->errors(), 'Data tidak valid.',  
        422);  
    }  
  
    $kategori = Kategori::create($request->only('nama'));
```

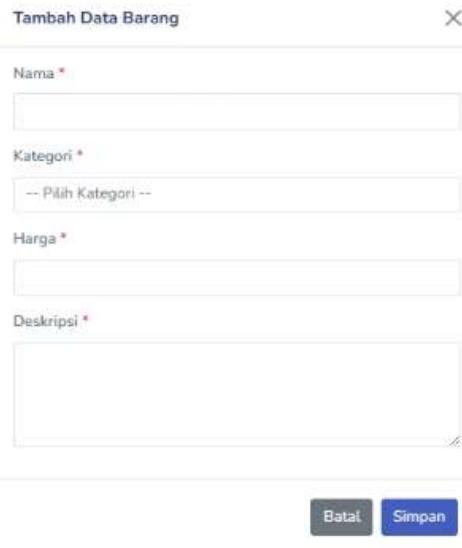
C. Membuat Form Master Barang



Tambah Data Barang

Nama *Kategori *Stok *Harga *Deskripsi *

Batal Simpan



Tambah Data Barang

Nama *Kategori * -- Pilih Kategori -- *Harga *Deskripsi *

Batal Simpan

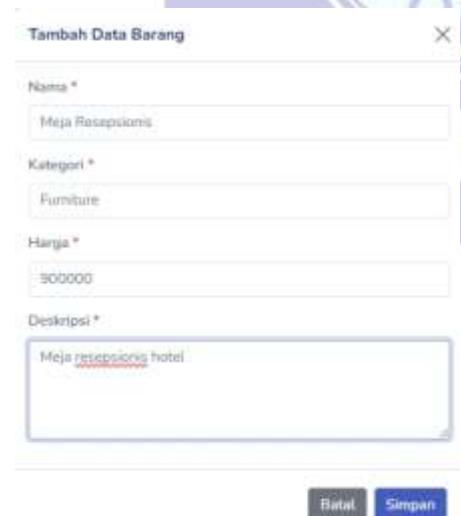
Gambar II. 14

Desain Form Master Barang

Gambar II. 15

Running Form Master Barang

1. Klik tombol tambah untuk tambah data Barang
2. Isi Form Master tambah data Barang



Tambah Data Barang

Nama * Meja Resepsionis *Kategori * Furniture *Harga * 500000 *Deskripsi * Meja resepsionis hotel *

Batal Simpan

Gambar II. 16

Tambah Data Barang



Tambah Data Barang

Nama *Kategori *Stok *Harga *Deskripsi *

Batal Simpan

Gambar II. 17

Form Master tambah_barang

Tabel II. 26

Spesifikasi properties komponen Form Master_Tambah_Barang

No	Komponen	Properti
1	Form input text 1	nama Teks : (kosongkan)
2	Form select 1	kategori Input : (kosongkan)
3	Form input text 3	harga Teks : (kosongkan)
4	Form input text 4	deskripsi Teks : (kosongkan)
5	Button 1	tombol_simpan Teks : Simpan
6	Button 2	tombol_batal Teks : Batal

Sumber : Hasil Penelitian (2024)

Logika Jalannya Program

- a. Masuk ke halaman admin barang.
- b. Klik tombol “Tambah” untuk menampilkan modal form data barang.
- c. Isi data kategori : nama, kategori, harga dan deskripsi.
- d. Jika ada data yang tidak sesuai, sistem memberikan validasi. Jika semua data valid, form disimpan ke database

3. Script pada tampilan halaman admin barang yang mana disini dapat melakukan data barang, tambah, edit dan hapus.

```
@extends('layouts.admin')

@section('title', 'Barang')

@push('style')
    <link rel="stylesheet" href="{{ asset('admin/extensions/datatables.net-bs5/css/dataTables.bootstrap5.min.css') }}">
    <link rel="stylesheet" href="{{ asset('admin/compiled/css/table-datable-jquery.css') }}">
    <link rel="stylesheet" href="{{ asset('admin/extensions/sweetalert2/sweetalert2.min.css') }}">
@endpush

@section('main')
    <div class="content-wrapper container">
        <div class="page-heading">
            <div class="row">
                <div class="col-12 col-md-6 order-md-1 order-last">
                    <h3>@yield('title')</h3>
                </div>
                <div class="col-12 col-md-6 order-md-2 order-first">
```

```

<nav aria-label="breadcrumb" class="breadcrumb-header float-
start float-lg-end">

    <ol class="breadcrumb">

        <li class="breadcrumb-item" href="/admin">Dashboard</li>

        <li class="breadcrumb-item active" current="page">@yield('title')</li>

    </ol>

```

</nav>

</div>

</div>

</div>

</div>

<div class="page-content">

<section class="row">

<div class="col-12">

<div class="card">

<div class="card-body">

<div class="d-flex justify-content-between align-items-center">

<h5 class="card-title mb-0">Data @yield('title')</h5>

<div>

<button class="btn btn-success btn-sm" onclick="getModal('createModal')">

<i class="bi bi-plus me-2"></i>Tambah

</button>

```

        </div>

        </div>

        <hr>

        <div class="table-responsive">

            <table class="table table-bordered table-striped" id="barang-
table" width="100%">

                <thead>

                    <tr>

                        <th width="5%">#</th>

                        <th>Nama</th>

                        <th>Kategori</th>

                        <th>Harga</th>

                        <th>Deskripsi</th>

                        <th width="15%">Aksi</th>

                    </tr>

                </thead>

                <tbody>

                    </tbody>

                </table>

            </div>

            </div>

            </div>

            </div>

            </div>

```

```
</div>

@include('pages.admin.barang.modal')

@endsection

@push('scripts')
<script src="{{ asset('admin/extensions/datatables.net/js/jquery.dataTables.min.js') }}"></script>
<script src="{{ asset('admin/extensions/datatables.net-bs5/js/dataTables.bootstrap5.min.js') }}"></script>
<script src="{{ asset('admin/extensions/sweetalert2/sweetalert2.min.js') }}"></script>
<script>
$(document).ready(function() {
    datatableCall('barang-table', '{{ route('admin.barang.index') }}', [
        { data: 'DT_RowIndex', name: 'DT_RowIndex' },
        { data: 'nama', name: 'nama' },
        { data: 'kategori', name: 'kategori' }
    ],
    {
        data: 'DT_RowIndex', name: 'DT_RowIndex'
    }
});
```

```
{  
    data: 'harga',  
    name: 'harga'  
},  
{  
    data: 'deskripsi',  
    name: 'deskripsi'  
},  
{  
    data: 'aksi',  
    name: 'aksi',  
}  
]);  
  
$("#saveData").submit(function(e) {  
    setButtonLoadingState("#saveData .btn.btn-primary", true);  
    e.preventDefault();  
    const kode = $("#saveData #id").val();  
  
    let url = "{{ route('admin.barang.store') }}";  
  
    const data = new FormData(this);  
  
    if (kode !== "") {  
        data.append("_method", "PUT");  
        url = `/admin/barang/${kode}`;  
    }  
});
```



```

const successCallback = function(response) {
    setButtonLoadingState("#saveData .btn.btn-primary", false);
    handleSuccess(response, "barang-table", "createModal");
};

const errorCallback = function(error) {
    setButtonLoadingState("#saveData .btn.btn-primary", false);
    handleValidationErrors(error, "saveData", ["nama", "harga",
    "kategori_id",
    "deskripsi"]);
};

ajaxCall(url, "POST", data, successCallback, errorCallback);
});

});

</script>
@endpush

```



4. Script pada tampilan modal yang mana disini berfungsi sebagai form tambah dan edit pada barang.

```

<div class="modal fade" id="createModal" tabindex="-1" aria-
labelledby="exampleModalLabel" aria-hidden="true"
data-bs-backdrop="static" data-bs-keyboard="false">

```

```

<div class="modal-dialog modal-dialog-centered">

    <div class="modal-content">

        <div class="modal-header">

            <h5 class="modal-title" id="exampleModalLabel"><span id="label-modal"></span> Data @yield('title')

            </h5>

            <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>

        </div>

        <form id="saveData" autocomplete="off">

            <div class="modal-body">

                <input type="hidden" id="id">

                <div class="form-group mb-3">

                    <label for="nama" class="form-label">Nama <span class="text-danger">*</span></label>

                    <input type="text" class="form-control" id="nama" name="nama">

                    <small class="invalid-feedback" id="errornama"></small>

                </div>

                <div class="form-group mb-3">

                    <label for="kategori_id" class="form-label">Kategori <span class="text-danger">*</span></label>

                    <select class="form-control" id="kategori_id" name="kategori_id">

                        <option value="">-- Pilih Kategori --</option>

```

```

@foreach ($kategori as $row)

    <option value="{{$row->id}}>{{$row->nama

}}</option>

@endforeach

</select>

<small class="invalid-feedback" id="errorkategori_id"></small>

</div>

<div class="form-group mb-3">

    <label for="harga" class="form-label">Harga <span class="text-
danger">*</span></label>

    <input type="number" class="form-control" id="harga"
name="harga">

    <small class="invalid-feedback" id="errorharga"></small>

</div>

<div class="form-group mb-3">

    <label for="deskripsi" class="form-label">Deskripsi <span
class="text-danger">*</span></label>

    <textarea type="text" class="form-control" id="deskripsi"
rows="4" name="deskripsi"></textarea>

    <small class="invalid-feedback" id="errordeskripsi"></small>

</div>

</div>

<div class="modal-footer">

    <button type="button" class="btn btn-secondary" data-bs-
dismiss="modal">Batal</button>

```

```

<button type="submit" class="btn btn-primary">Simpan</button>

</div>

</form>

</div>

</div>

```

5. Script pada proses tambah data barang disini ada beberapa validasi untuk data yang diinput untuk dapat mendapatkan data yang valid dan disimpan pada database.

```

public function store(Request $request)
{
    $validator = Validator::make($request->all(), [
        'kategori_id' => 'required|exists:kategori,id',
        'nama' => 'required|string',
        'deskripsi' => 'required|string',
        'harga' => 'required|integer',
    ]);

    if ($validator->fails()) {
        return $this->errorResponse($validator->errors(), 'Data tidak valid.', 422);
    }

    $barang = Barang::create($request->only('kategori_id', 'nama', 'deskripsi',
        'harga'));
}

```

```
return $this->successResponse($barang, 'Data Barang ditambahkan.');
```

}



UNIVERSITAS

D. Membuat Form Master Barang Gambar



Gambar II. 18



Gambar II. 19

Desain Form Master Barang Gambar

Running Form Master Barang Gambar

1. Klik tombol tambah untuk tambah data Barang Gambar
2. Isi Form Master tambah data Barang Gambar



Gambar II. 20

Tambah Data Barang Gambar



Gambar II. 21

Form Master barang_gambar

Tabel II. 27

Spesifikasi properties komponen Form Master_Tambah_Barang_Gambar

No	Komponen	Properti
1	Form input file	gambar File : (kosongkan)
5	Button 1	tombol_simpan Teks : Simpan
6	Button 2	tombol_batal Teks : Batal

Sumber : Hasil Penelitian (2024)

Logika Jalannya Program

- a. Masuk ke halaman admin barang gambar.
- b. Klik tombol “Tambah” untuk menampilkan modal form data barang gambar.
- c. Isi data galeri: gambar.
- d. Jika ada data yang tidak sesuai, sistem memberikan validasi. Jika semua data valid, form disimpan ke database

3. Script pada tampilan modal detail barang yang mana disini berfungsi sebagai form tambah dan edit pada data gambar barang.

```
<div class="modal fade" id="createModal" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true" data-bs-backdrop="static" data-bs-keyboard="false">

<div class="modal-dialog modal-dialog-centered">

    <div class="modal-content">
```

```

<div class="modal-header">

    <h5 class="modal-title" id="exampleModalLabel"><span id="label-
modal"></span> Data @yield('title')

    </h5>

    <button type="button" class="btn-close" data-bs-dismiss="modal" aria-
label="Close"></button>

</div>

<form id="saveData" autocomplete="off">

    <div class="modal-body">

        <input type="hidden" id="id">

        <div class="form-group">

            <label for="gambar" class="form-label">Gambar <span
class="text-danger">*</span></label>

            <input type="file" name="gambar" id="gambar" class="dropify"
data-height="200">

            <small class="text-danger" id="errorgambar"></small>
        </div>

    </div>

    <div class="modal-footer">

        <button type="button" class="btn btn-secondary" data-bs-
dismiss="modal">Batal</button>

        <button type="submit" class="btn btn-primary">Simpan</button>

    </div>

</form>

</div>

```

```
</div>
```

```
</div>
```

4. Script pada proses tambah data gambar barang disini ada beberapa validasi untuk data yang diinput untuk dapat mendapatkan data yang valid dan disimpan pada database

```
public function store(Request $request)
```

```
{
```

```
    $validator = Validator::make($request->all(), [  
        'gambar' => 'required|image|mimes:jpg,jpeg,png',  
    ]);  
  
    if ($validator->fails()) {  
        return $this->errorResponse($validator->errors(), 'Data tidak valid.', 422);  
    }
```

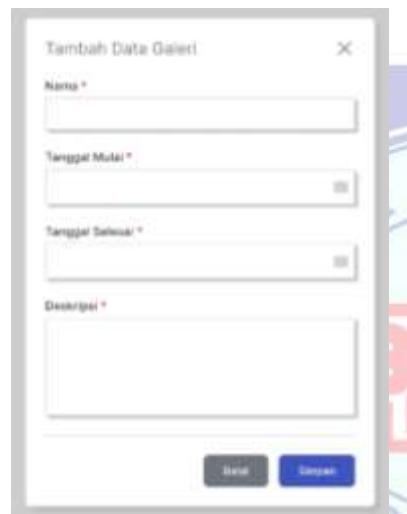
```
    if ($request->hasFile('gambar')) {  
  
        $file = $request->file('gambar')->hashName();  
  
        $request->file('gambar')->storeAs('public/galeri/barang', $file);  
    }
```

```
$galeri = BarangGambar::create([  
    'barang_id' => $request->barang_id,  
    'gambar' => $file,  
]);
```

```
return $this->successResponse($galeri, 'Data Barang Gambar  
ditambahkan.');
```

}

E. Membuat Form Master Galeri



The image shows the design of a 'Tambah Data Galeri' (Add Gallery Data) form. It consists of four input fields: 'Nama *' (Name), 'Tanggal Mulai *' (Start Date), 'Tanggal Selesai *' (End Date), and 'Deskripsi *' (Description). Below the input fields are two buttons: 'Batal' (Cancel) and 'Simpan' (Save).

Gambar II. 22

Desain Form Master Galeri



The image shows the running state of the 'Tambah Data Galeri' form. The input fields ('Nama *', 'Tanggal Mulai *', 'Tanggal Selesai *', 'Deskripsi *') are populated with sample data: 'nametest', '2019/01/01', '2019/01/02', and 'tes'. The 'Batal' and 'Simpan' buttons are visible at the bottom.

Gambar II. 23

Running Form Master Galeri

1. Klik tombol tambah untuk tambah data Galeri
2. Isi Form Master tambah data Galeri

Gambar II. 24

Tambah Data Galeri

Gambar II. 25

Form Master tambah_galeri

Tabel II. 28

Spesifikasi properties komponen Form Master_Tambah_Galeri

No	Komponen	Properti
1	Form input text 1	nama Teks : (kosongkan)
2	Form input date 1	tanggal_mulai Input : (kosongkan)
3	Form input date 2	tanggal_selesai Input : (kosongkan)
4	Form input text 2	deskripsi Teks : (kosongkan)
5	Button 1	tombol_simpan Teks : Simpan

6	Button 2	tombol_batal Teks : Batal
---	----------	------------------------------

Sumber : Hasil Penelitian (2024)

Logika Jalannya Program

- a. Masuk ke halaman admin galeri.
- b. Klik tombol “Tambah” untuk menampilkan modal form data barang.
- c. Isi data galeri: nama, kategori, tanggal mulai, tanggal selesai dan deskripsi.
- d. Jika ada data yang tidak sesuai, sistem memberikan validasi. Jika semua data valid, form disimpan ke database

- 3. Script pada tampilan halaman admin galeri yang mana disini dapat melakukan data galeri, tambah, edit dan hapus.

```

@extends('layouts.admin')

@section('title', 'Galeri')

@push('style')
<link rel="stylesheet" href="{{ asset('admin/extensions/datatables.net-bs5/css/dataTables.bootstrap5.min.css') }}>
<link rel="stylesheet" href="{{ asset('admin/compiled/css/table-datable-jquery.css') }}>
<link rel="stylesheet" href="{{ asset('admin/extensions/sweetalert2/sweetalert2.min.css') }}>

@endpush

```

```

@section('main')

<div class="content-wrapper container">

    <div class="page-heading">

        <div class="page-title">

            <div class="row">

                <div class="col-12 col-md-6 order-md-1 order-last">

                    <h3>@yield('title')</h3>

                </div>

                <div class="col-12 col-md-6 order-md-2 order-first">

                    <nav aria-label="breadcrumb" class="breadcrumb-header
float-start float-lg-end">

                        <ol class="breadcrumb">

                            <li class="breadcrumb-item" href="/admin">Dashboard</li>

                            <li class="breadcrumb-item active" aria-
current="page">@yield('title')</li>

                        </ol>

                    </nav>

                </div>

            </div>

        </div>

        <div class="page-content">

            <section class="row">

```

```

<div class="col-12">
    <div class="card">
        <div class="card-body">
            <div class="d-flex justify-content-between align-items-center">
                <h5 class="card-title mb-0">Data @yield('title')</h5>
                <div>
                    <button class="btn btn-success btn-sm" onclick="getModal('createModal')">
                        <i class="bi bi-plus me-2"></i>Tambah
                    </button>
                </div>
            </div>
            <hr>
            <div class="table-responsive">
                <table class="table table-bordered table-striped" id="galeri-table" width="100%">
                    <thead>
                        <tr>
                            <th width="5%">#</th>
                            <th>Nama</th>
                            <th>Tanggal Mulai</th>
                            <th>Tanggal Selesai</th>
                            <th>Deskripsi</th>
                            <th width="15%">Aksi</th>
                        </tr>
                    </thead>

```



datatableCall('galeri-table', '{{ route('admin.galeri.index') }}', [{
 data: 'DT_RowIndex',
 name: 'DT_RowIndex'
},
{
 data: 'nama',
 name: 'nama'
},
{
 data: 'tanggal_mulai',
 name: 'tanggal_mulai'
},
{
 data: 'tanggal_selesai',
 name: 'tanggal_selesai'
},
{
 data: 'deskripsi',
 name: 'deskripsi'
},
{
 data: 'aksi',
 name: 'aksi',
}
]);

```
$("#saveData").submit(function(e) {  
    setButtonLoadingState("#saveData .btn.btn-primary", true);  
    e.preventDefault();  
  
    const kode = $("#saveData #id").val();  
  
    let url = "{{ route('admin.galeri.store') }}";  
  
    const data = new FormData(this);
```

```
if (kode !== "") {  
  
    data.append("_method", "PUT");  
  
    url = `/admin/galeri/${kode}`;  
}  
  
const successCallback = function(response) {  
  
    setButtonLoadingState("#saveData .btn.btn-primary", false);  
  
    handleSuccess(response, "galeri-table", "createModal");  
};
```

```
const errorCallback = function(error) {  
  
    setButtonLoadingState("#saveData .btn.btn-primary", false);  
  
    handleValidationErrors(error, "saveData", ["nama",  
        "deskripsi", "tanggal_mulai", "tanggal_selesai"  
    ]);  
};
```

```

        ajaxCall(url, "POST", data, successCallback, errorCallback);

    });

});

</script>

@endpush

```

4. Script pada tampilan modal yang mana disini berfungsi sebagai form tambah dan edit pada galeri.

```

<div class="modal fade" id="createModal" tabindex="-1" aria-
labelledby="exampleModalLabel" aria-hidden="true"
data-bs-backdrop="static" data-bs-keyboard="false">
<div class="modal-dialog modal-dialog-centered">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title" id="exampleModalLabel"><span
id="label-modal"></span> Data @yield('title')
</h5>
<button type="button" class="btn-close" data-bs-dismiss="modal"
aria-label="Close"></button>
</div>
<form id="saveData" autocomplete="off">
<div class="modal-body">
<input type="hidden" id="id">
<div class="form-group mb-3">

```

```

<label for="nama" class="form-label">Nama <span
class="text-danger">*</span></label>

<input type="text" class="form-control" id="nama"
name="nama">

<small class="invalid-feedback" id="errornama"></small>

</div>

<div class="form-group mb-3">

<label for="tanggal_mulai" class="form-label">Tanggal Mulai
Tanggal
Selesai <span
class="text-danger">*</span></label>

<input type="date" class="form-control" id="tanggal_selesai"
name="tanggal_selesai">
<small
class="invalid-feedback"
id="errortanggal_selesai"></small>

</div>

<div class="form-group mb-3">

```

```

<label for="deskripsi" class="form-label">Deskripsi <span
class="text-danger">*</span></label>

<textarea type="text" class="form-control" id="deskripsi"
rows="4" name="deskripsi"></textarea>

<small class="invalid-feedback" id="errordeskripsi"></small>

</div>

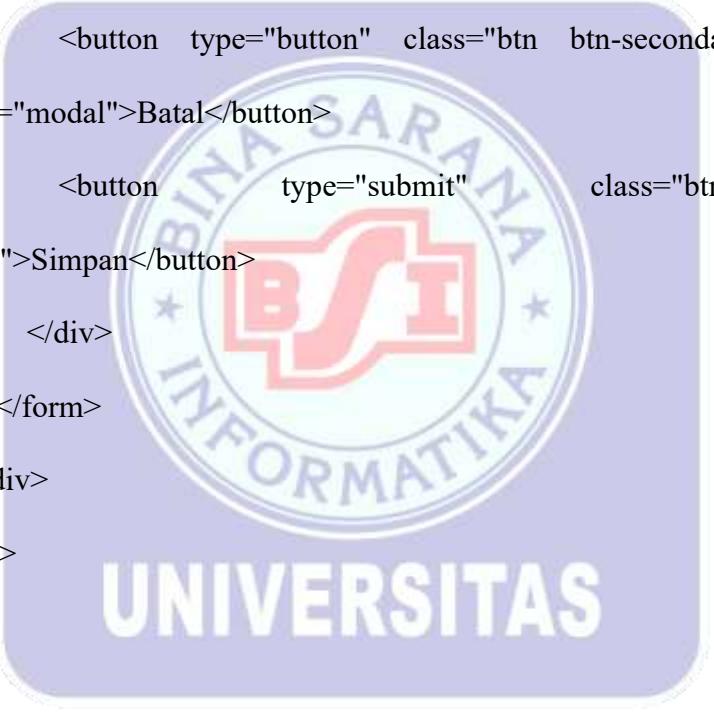
</div>

<div class="modal-footer">

    <button type="button" class="btn btn-secondary" data-bs-
dismiss="modal">Batal</button>

    <button type="submit" class="btn btn-
primary">Simpan</button>

```



```

        </div>
    </form>
</div>
</div>
</div>

```

5. Script pada proses tambah data galeri disini ada beberapa validasi untuk data yang diinput untuk dapat mendapatkan data yang valid dan disimpan pada database

```

public function store(Request $request)
{
    $validator = Validator::make($request->all(), [
        'nama' => 'required|string',

```

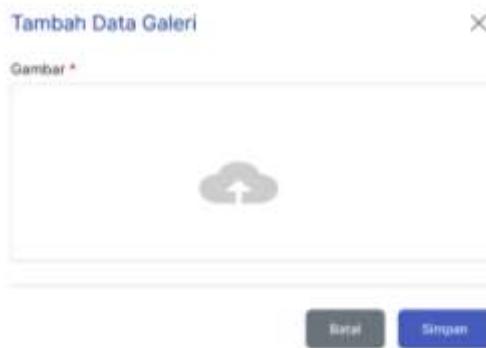
```
'tanggal_mulai' => 'required|date',
'tanggal_selesai' => 'required|date|after_or_equal:tanggal_mulai',
'deskripsi' => 'required|string',
]);

if ($validator->fails()) {
    return $this->errorResponse($validator->errors(), 'Data tidak valid.', 422);
}

$galeri = Galeri::create($request->only('nama', 'tanggal_mulai', 'tanggal_selesai', 'deskripsi'));
return $this->successResponse($galeri, 'Data Galeri ditambahkan.');
}
```



F. Membuat Form Master Galeri Gambar



Tambah Data Galeri

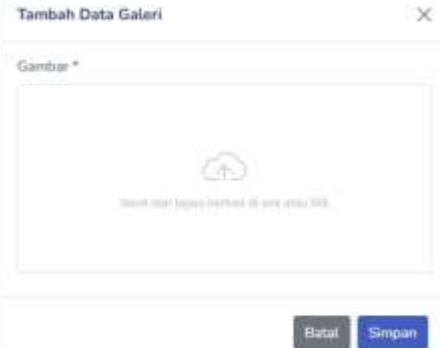
Gambar *

Batal Simpan

A screenshot of the 'Tambah Data Galeri' (Add Gallery Data) form. It features a large input field for uploading images, represented by a cloud icon with an upward arrow. Below the input field are two buttons: 'Batal' (Cancel) and 'Simpan' (Save).

Gambar II. 26

Desain Form Master Galeri Gambar



Tambah Data Galeri

Gambar *

Batal Simpan

A screenshot of the same 'Tambah Data Galeri' form, but it now displays a message: 'Silakan upload berkas gambar di sini atau link'. This indicates that the form is active and ready for user input.

Gambar II. 27

Running Form Master Galeri Gambar

1. Klik tombol tambah untuk tambah data Galeri Gambar
2. Isi Form Master tambah data Galeri Gambar



Tambah Data Galeri

Gambar *

Batal Simpan

A screenshot of the 'Tambah Data Galeri' form showing a successfully uploaded image of a large, colorful mural on a wall. The image is displayed in the input field. At the bottom of the form are the 'Batal' (Cancel) and 'Simpan' (Save) buttons.

Gambar II. 28

Tambah Data Galeri Gambar



Tambah Data Galeri

Gambar *

Batal Simpan

A screenshot of the 'Tambah Data Galeri' form with a green circular overlay highlighting the 'Simpan' (Save) button at the bottom right corner.

Gambar II. 29

Form Master tambah_galeri_gambar

Tabel II. 29

Spesifikasi properties komponen Form Master_Tambah_Galeri_Gambar

No	Komponen	Properti
1	Form input file	gambar File : (kosongkan)
2	Button 1	tombol_simpan Teks : Simpan
3	Button 2	tombol_batal Teks : Batal

Sumber : Hasil Penelitian (2024)

Logika Jalannya Program

- a. Masuk ke halaman admin galeri gambar.
 - b. Klik tombol “Tambah” untuk menampilkan modal form data galeri gambar.
 - c. Isi data galeri: gambar.
 - d. Jika ada data yang tidak sesuai, sistem memberikan validasi. Jika semua data valid, form disimpan ke database
3. Script pada tampilan modal detail galeri yang mana disini berfungsi sebagai form tambah dan edit pada data gambar galeri.

```
<div class="modal fade" id="createModal" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true" data-bs-backdrop="static" data-bs-keyboard="false">
<div class="modal-dialog modal-dialog-centered">
<div class="modal-content">
```

```
<div class="modal-header">  
    <h5 class="modal-title" id="exampleModalLabel"><span id="label-modal"></span> Data @yield('title')  
    </h5>  
    <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>  
</div>  
<form id="saveData" autocomplete="off">  
    <div class="modal-body">  
        <input type="hidden" id="id">  
        <div class="form-group">  
            <label for="gambar" class="form-label">Gambar <span class="text-danger">*</span></label>  
            <input type="file" name="gambar" id="gambar" class="dropify" data-height="200">  
            <small class="text-danger" id="errorgambar"></small>  
        </div>  
    </div>  
    <div class="modal-footer">  
        <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Batal</button>  
        <button type="submit" class="btn btn-primary">Simpan</button>  
    </div>  
</form>  
</div>
```

```
</div>
```

```
</div>
```

4. Script pada proses tambah data galeri di detail galeri disini ada beberapa validasi untuk data yang diinput untuk dapat mendapatkan data yang valid dan disimpan pada database

```
public function store(Request $request)
```

```
{
```

```
$validator = Validator::make($request->all(), [  
    'gambar' => 'required|image|mimes:jpg,jpeg,png',  
]);
```

```
if ($validator->fails()) {
```

```
    return $this->errorResponse($validator->errors(), 'Data tidak valid.',  
        422);
```

```
}
```



UNIVERSITAS

```
if ($request->hasFile('gambar')) {
```

```
    $file = $request->file('gambar')->hashName();
```

```
    $request->file('gambar')->storeAs('public/galeri/gambar', $file);
```

```
}
```

```
$galeri = GaleriGambar::create([
```

```
    'galeri_id' => $request->galeri_id,
```

```
    'gambar' => $file,
```

]);

```
return $this->successResponse($galeri, 'Data Galeri Gambar  
ditambahkan.');
```

}

G. Membuat Form Master Pengaturan

Gambar II. 30

Gambar II. 31

Desain Form Master Pengaturan

Running Form Master Pengaturan

1. Klik tombol sidebar Pengaturan
2. Ubah Form Master Pengaturan

Data Pengaturan

Nama *
Esa Gallery

Email *
wida@gmail.com

No. HP *
+6282126056028

Alamat *
Jl. Nasional 3, Sukamanah 04/02, sindangkasih, ciamis

Deskripsi
Esa Gallery

Simpan

Data Pengaturan

1 Nama *
2 Email *
3 No Hp *
4 Alamat *
5 Deskripsi *
6 Simpan

Gambar II. 32

Edit Dat Pengaturan

Gambar II. 33

Form Master ubah_pengaturan

Tabel II. 30

Spesifikasi properties komponen Form Master_Pengaturan

No	Komponen	Properti
1	Form input text 1	nama Teks : (kosongkan)
2	Form input text 2	email Teks : (kosongkan)
3	Form input text 3	no_hp Teks : (kosongkan)
4	Form input text 4	alamat Teks : (kosongkan)
5	Form input text 5	deskripsi

		Teks : (kosongkan)
6	Button 1	tombol_simpan Teks : Simpan

Sumber : Hasil Penelitian (2024)

Logika Jalannya Program

- a. Masuk ke halaman admin pengaturan yang menampilkan form pengaturan.
- b. Ubah data pengaturan : nama, email, no hp, alamat dan deskripsi.
- c. Jika ada data yang tidak sesuai, sistem memberikan validasi. Jika semua data valid, form diubah pada database.
- 3. Script pada tampilan halaman admin pengaturan yang mana disini dapat melakukan ubah data pengaturan.

```
@extends('layouts.admin')

@section('title', 'Pengaturan')

@push('style')
<link rel="stylesheet" href="{{ asset('admin/extensions/sweetalert2/sweetalert2.min.css') }}"/>
@endpush


```

```
@section('main')
<div class="content-wrapper container">
<div class="page-heading">
```

```

<div class="page-title">
    <div class="row">
        <div class="col-12 col-md-6 order-md-1 order-last">
            <h3>@yield('title')</h3>
        </div>
        <div class="col-12 col-md-6 order-md-2 order-first">
            <nav aria-label="breadcrumb" class="breadcrumb-header float-
                start float-lg-end">
                <ol class="breadcrumb">
                    <li class="breadcrumb-item"><a href="/admin">Dashboard</a></li>
                    <li class="breadcrumb-item active" aria-
                        current="page">@yield('title')</li>
                </ol>
            </nav>
        </div>
    </div>
</div>

<div class="page-content">
    <section class="row">
        <div class="col-12 col-lg-6">
            <div class="card">
                <div class="card-header">
                    <h4 class="text-dark">Data @yield('title')</h4>
                </div>

```

```

</div>

<div class="card-body">

    <form id="updateData">

        @method('PUT')

        <div class="form-group mb-3">

            <label for="nama" class="form-label">Nama <span
                class="text-danger">*</span></label>

            <input type="text" class="form-control" id="nama"
                name="nama"
                value="{{ $pengaturan->nama }}"

            <small class="invalid-feedback"
                id="errornama"></small>
        </div>

        <div class="form-group mb-3">

            <label for="email" class="form-label">Email <span
                class="text-danger">*</span></label>

            <input type="email" class="form-control" id="email"
                name="email"
                value="{{ $pengaturan->email }}"

            <small class="invalid-feedback"
                id="erroremail"></small>
        </div>

        <div class="form-group mb-3">

            <label for="no_hp" class="form-label">No. HP <span
                class="text-danger">*</span></label>

```



```

<input type="text" class="form-control" id="no_hp"
name="no_hp"
value="{{ $pengaturan->no_hp }}>
<small class="invalid-feedback">
<small id="errorno_hp"></small>
</div>
<div class="form-group mb-3">
<label for="alamat" class="form-label">Alamat <span
class="text-danger">*</span></label>
<textarea name="alamat" id="alamat" class="form-
control" rows="4">{{ $pengaturan->alamat }}</textarea>
<small class="invalid-feedback">
<small id="erroralamat"></small>
</div>
<div class="form-group mb-3">
<label for="deskripsi" class="form-
label">Deskripsi</label>
<textarea class="form-control" id="deskripsi"
name="deskripsi" rows="4">{{ $pengaturan->deskripsi }}</textarea>
<small class="invalid-feedback">
<small id="errordeskripsi"></small>
</div>
<div class="form-group">
<button type="submit" class="btn btn-
primary">Simpan</button>

```

```

        </div>

        </form>

        </div>

        </div>

        </div>

        </section>

        </div>

        </div>

@endsection

@push('scripts')

<script src="{{ asset('admin/extensions/sweetalert2/sweetalert2.min.js') }}"></script>

<script>

$(document).ready(function() {

    $("#updateData").submit(function(e) {

        setButtonLoadingState("#updateData .btn.btn-primary", true);

        e.preventDefault();

        const url = `{{ route('admin.pengaturan') }}`;

        const data = new FormData(this);

        const successCallback = function(response) {

            setButtonLoadingState("#updateData .btn.btn-primary", false);

            handleSuccess(response, null, null, "no");
        }
    });
});

```

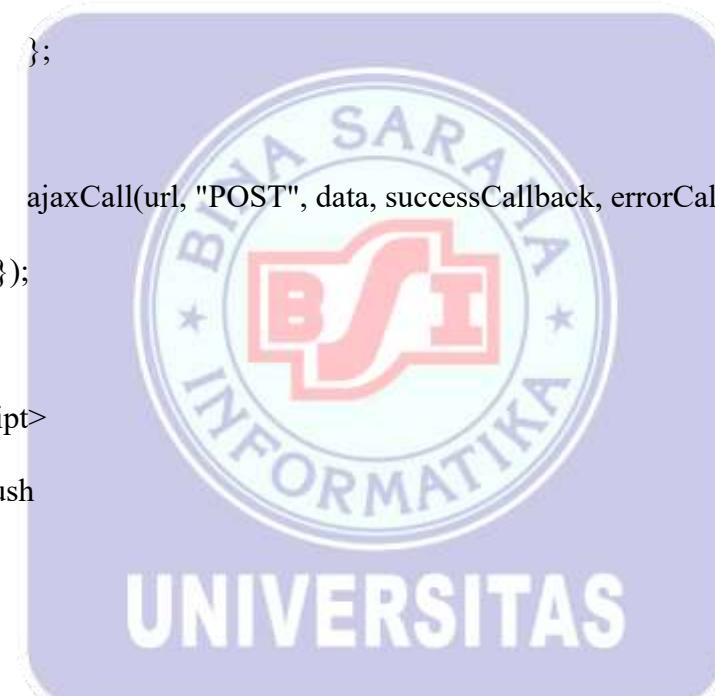
```
};

const errorCallback = function(error) {
    setButtonLoadingState("#updateData .btn.btn-primary", false);
    handleValidationErrors(error, "updateData", ["nama", "email",
    "no_hp", "alamat",
    "deskripsi"
]);
};

ajaxCall(url, "POST", data, successCallback, errorCallback);
});

});

</script>
@endpush
```



H. Membuat Form Master Profile

The image shows the design of the Master Profile form. It consists of two separate forms: 'Data Profile' and 'Ubah Password'.
The 'Data Profile' form has fields for 'Nama *' (Name) and 'Email *' (Email), both with placeholder text. A blue 'Simpan' (Save) button is at the bottom.
The 'Ubah Password' form has fields for 'Password Lama *' (Old Password), 'Password Baru *' (New Password), and 'Konfirmasi password *' (Confirm Password), all with placeholder text. A blue 'Simpan' (Save) button is at the bottom.

Gambar II. 34
Desain Form Master Profile

The image shows the running state of the Master Profile form, specifically the 'Data Profile' section. The 'Nama' field contains 'Administrator' and the 'Email' field contains 'wida@gmail.com'. A blue 'Simpan' (Save) button is visible at the bottom.

Gambar II. 35
Running Form Master Profile

1. Klik tombol sidebar Profile
2. Ubah Form Master Profile

Data Profile

Nama *
Admin

Email *
wida@gmail.com

Simpan

Ubah Password

Password Lama *

Password Baru *

Konfirmasi Password *

Simpan

Data Profile

Nama *
1

Email *
2

Simpan

Ubah Password

Password Lama *
3

Password Baru *
4

Konfirmasi password *
3

Simpan

Gambar II. 36

Ubah Profile

Gambar II. 37

Form Master ubah_profile

Tabel II. 31

Spesifikasi properties komponen Form Master_Profile

No	Komponen	Properti
1	Form input text 1	nama Teks : (kosongkan)
2	Form input text 2	email Teks : (kosongkan)
3	Button 1	tombol_simpan Teks : Simpan

Sumber : Hasil Penelitian (2024)

Tabel II. 32

Spesifikasi properties komponen Form Master_Ubah_Password

No	Komponen	Properti
1	Form input text 1	password lama Teks : (kosongkan)
2	Form input text 2	password baru Teks : (kosongkan)
3	Form input text	konfirmasi password Teks : Simpan
4	Button 1	tombol_simpan Teks : Simpan

Sumber : Hasil Penelitian (2024)

Logika Jalannya Program

- a. Masuk ke halaman admin pengaturan yang menampilkan form pengaturan.
 - b. Ubah data pengaturan : nama, email, no hp, alamat dan deskripsi.
 - c. Jika ada data yang tidak sesuai, sistem memberikan validasi. Jika semua data valid, form diubah pada database.
 - d. Script pada tampilan halaman admin profile yang mana disini dapat melakukan ubah data profile.
-
3. Script pada tampilan halaman admin profile yang mana disini dapat mengubah data profile dan ubah password.

@extends('layouts.admin')

```

@section('title', 'Profile')

@push('style')
<link rel="stylesheet" href="{{ asset('admin/extensions/sweetalert2/sweetalert2.min.css') }}"
@endpush

@section('main')
<div class="content-wrapper container">
<div class="page-heading">
<div class="page-title">
<div class="row">
<div class="col-12 col-md-6 order-md-1 order-last">
<h3>@yield('title')</h3>
</div>
<div class="col-12 col-md-6 order-md-2 order-first">
<nav aria-label="breadcrumb" class="breadcrumb-header float-start float-lg-end">
<ol class="breadcrumb">
<li class="breadcrumb-item"><a href="/admin">Dashboard</a></li>
<li class="breadcrumb-item active" aria-current="page">@yield('title')</li>
</ol>
</nav>

```

```

</div>

</div>

</div>

<div class="page-content">

<section class="row">

<div class="col-12 col-lg-6">

<div class="card">

<div class="card-header">

<h4 class="text-dark">Data @yield('title')</h4>

</div>

<div class="card-body">

<form id="updateData" @method('PUT')

<div class="form-group mb-3">

<label for="nama" class="form-label">Nama <span class="text-danger">*</span></label>

<input type="text" class="form-control" id="nama" name="nama" value="{{$auth->user()->nama}}>

<small class="invalid-feedback" id="errornama"></small>

</div>

<div class="form-group mb-3">

<label for="email" class="form-label">Email <span

```

```

        class="text-danger">>*</span></label>

<input type="email" class="form-control" id="email"
name="email"

value="{{ auth()->user()->email }}>

<small class="invalid-feedback"
id="erroremail"></small>

</div>

<div class="form-group">

<button type="submit" class="btn btn-primary">Simpan</button>

</div>

</form>

</div>

</div>

<div class="card">

<div class="card-header">
    <h4 class="text-dark">Ubah Password</h4>
</div>

<div class="card-body">

<form id="updatePassword">

    @method('PUT')

    <div class="form-group mb-3">

        <label for="password_lama" class="form-label">Password Lama <span
class="text-danger">*</span></label>

```

```

<input type="password" class="form-control"
id="password_lama" name="password_lama">
<small class="invalid-feedback"
id="errorpassword_lama"></small>
</div>
<div class="form-group mb-3">
<label for="password" class="form-label">Password
Baru <span
class="text-danger">*</span></label>
<input type="password" class="form-control"
id="password" name="password">
<small class="invalid-feedback"
id="errorpassword"></small>
</div>
<div class="form-group mb-3">
<label for="password_confirmation" class="form-
label">Konfirmasi Password <span
class="text-danger">*</span></label>
<input type="password" class="form-control"
id="password_confirmation"
name="password_confirmation">
<small class="invalid-feedback"
id="errorpassword_confirmation"></small>
</div>
<div class="form-group">

```

```

        <button type="submit" class="btn btn-primary">Simpan</button>
    </div>
</form>
</div>
</div>
</div>
</section>
</div>
</div>
@endsection
@push('scripts')
<script src="{!! asset('admin/extensions/sweetalert2/sweetalert2.min.js') !!}"></script>
<script>
$(document).ready(function() {
    $("#updateData").submit(function(e) {
        setButtonLoadingState("#updateData .btn.btn-primary", true);
        e.preventDefault();
        const url = `{{ route('admin.profile') }}`;
        const data = new FormData(this);
        const successCallback = function(response) {
            setButtonLoadingState("#updateData .btn.btn-primary", false);
        }
    })
})

```



```

        handleSuccess(response, null, null, "no");

    };

    const errorCallback = function(error) {
        setButtonLoadingState("#updateData .btn.btn-primary", false);
        handleValidationErrors(error, "updateData", ["nama", "email"]);
    };

    ajaxCall(url, "POST", data, successCallback, errorCallback);
});

$("#updatePassword").submit(function(e) {
    setButtonLoadingState("#updatePassword .btn.btn-primary", true);
    e.preventDefault();
    const url = `{{ route('admin.profile.password') }}`;
    const data = new FormData(this);
    const successCallback = function(response) {
        setButtonLoadingState("#updatePassword .btn.btn-primary",
            false);
        handleSuccess(response, null, null, "no");
        $('#updatePassword .form-control').removeClass("is-
invalid").val("");
        $('#updatePassword .invalid-feedback').html("");
    };
});

```

```

    const errorCallback = function(error) {
        setButtonLoadingState("#updatePassword" .btn.btn-primary",
        false);
        handleValidationErrors(error, "updatePassword",
        ["password_lama", "password"]);
    };

```

```

        ajaxCall(url, "POST", data, successCallback, errorCallback);
    });
});
</script>
@endpush

```

4. Script pada proses ubah password di profile disini ada validasi untuk data yang diinput untuk dapat mendapatkan data yang valid dan disimpan pada database

public function updatePassword(Request \$request)

```

{
    $validator = Validator::make($request->all(), [
        'password_lama' => 'required|min:8',
        'password' => 'required|min:8|confirmed',
    ]);
}
```

```

if ($validator->fails()) {
```

```
        return $this->errorResponse($validator->errors(), 'Data tidak valid.',  
422);  
    }  
  
    $user = auth()->user();  
  
    if (!$user) {  
        return $this->errorResponse(null, 'Data Profil tidak ditemukan.', 404);  
    }  
  
    if (!Hash::check($request->password_lama, $user->password)) {  
        return $this->errorResponse(null, 'Password lama tidak sesuai.', 422);  
    }  
  
    $user->update($request->only('password'));  
  
    return $this->successResponse($user, 'Data password diubah.');
```

2.4 Rancangan Transaksi

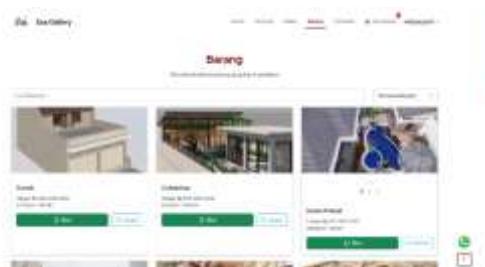
A. Membuat Form Transaksi User

Desain Transaksi User



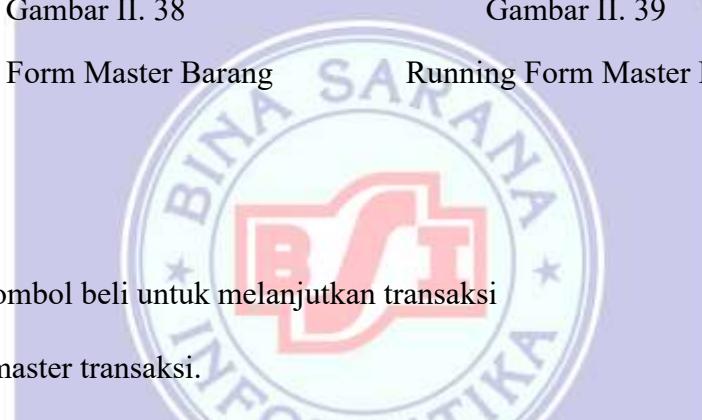
Gambar II. 38

Running Transaksi User



Gambar II. 39

Desain Form Master Barang



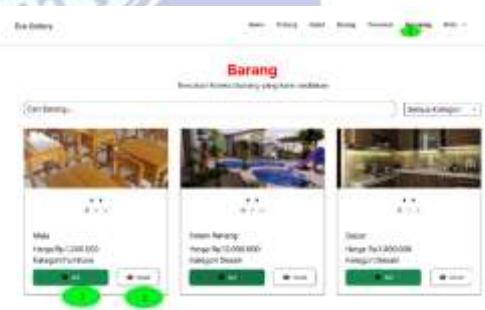
Running Form Master Barang

1. Klik Tombol beli untuk melanjutkan transaksi
2. Form master transaksi.



Gambar II. 40

Masukan Keranjang



Gambar II. 41

Form_master_masukkan_keranjang

Tabel II. 33

Spesifikasi properties komponan form_master_transaksi

1	Button 1	Tombol_Beli Teks : Beli
2	Button 2	Tombol_Detail Teks : Detail

Sumber : Hasil Penelitian (2024)

Logika Jalannya Program

- a. Masuk ke halaman Barang
 - b. Klik tombol “Detail” Untuk Melihat Detail Barang
 - c. Klik Tombol “Beli” Untuk Membeli Barang
 - d. Jika sudah klik tombol beli maka selanjutnya masuk ke halaman keranjang
3. Script pada tampilan halaman barang yang mana disini dapat melakukan transaksi.

@forelse (\$barang as \$row)

```
<div class="col-lg-4 col-12 col-md-6 mb-3">  
  <div class="card shadow-sm rounded-3 h-100">  
    <div class="card-body p-0">  
      <div class="mb-3">  
        <div class="owl-carousel owl-theme">  
          @forelse ($row->barangGambars as $barangGambars)  
            <div class="item">
```

```

nama }}"
      >
</div>

@empty
<div class="item">
    nama }}"
        >
</div>
@endforelse
</div>
</div>
<div class="p-3">
    <div class="d-flex justify-content-between">
        <div>
            <p class="card-title fw-bold">{{ $row->nama }}</p>
            <div class="mb-3 d-flex flex-column">
                <small>Harga: {{ formatRupiah($row->harga) }}</small>
                <small>Kategori : {{ $row->kategori->nama }}</small>
            </div>
        </div>
    </div>
    <div class="d-flex justify-content-between gap-2">
        @auth

```

```

<button onclick="addCart({{ $row->id }})" class="btn btn-
success d-block w-100"><i
    class="bi bi-cart me-2"></i>Beli</button>

<a href="/barang/{{ $row->id }}" class="btn btn-outline-info
d-flex"><i
    class="bi bi-eye me-2"></i>Detail</a>

@endif

<a href="/barang/{{ $row->id }}" class="btn btn-outline-info
d-block w-100"><i
    class="bi bi-eye me-2"></i>Detail</a>

@endauth

</div>
</div>
</div>
</div>
</div>
</div>
@empty
<div class="col-12">
    <div class="text-center my-5 py-5">
        <div class="fw-semibold">Barang Tidak Ditemukan</div>
    </div>
</div>
@endif
<div class="col-12">
    <div class="d-flex justify-content-center">

```

```

{!! $barang->links() !!}
</div>
</div>

```

B.Membuat Form Transaksi Keranjang

Desain Transaksi Keranjang

Gambar II. 42

Running Transaksi Keranjang

Gambar II. 43

Desain Form Master Profile

Running Form Master Profile



1. Klik tombol bayar untuk melanjutkan transaksi
2. Form Master Keranjang

Gambar II. 44

Keranjang

Gambar II. 45

Form_master_keranjang

Tabel II. 34

Spesifikasi properties komponan form_master_transaksi

1	Button 1	Tombol_Edit
2	Button 2	Tombol_Hapus
3.	Button 3	Tombol Bayar Teks : Bayar

Sumber : Hasil Penelitian (2024)

Logika Jalannya Program

- a. Masuk ke halaman Keranjang
 - b. Klik tombol “Edit” untuk menampilkan modal form data edit quantity
 - c. Klik icon “hapus” untuk menghapus
 - d. Klik tombol “bayar” untuk membayar
 - e. Selanjutnya jika sudah membayar maka akan muncul kode transaksi
3. Script pada tampilan halaman keranjang yang mana disini dapat melakukan edit quantity, hapus barang dan bayar.

```
@extends('layouts.frontend')
```

```
@section('title', 'Keranjang')
```

```
@push('style')
```

```
<link rel="stylesheet" href="{{
```

```
asset('admin/extensions/sweetalert2/sweetalert2.min.css') }}>
```

```

<link rel="stylesheet" href="{{ asset('admin/extensions/datatables.net-bs5/css/dataTables.bootstrap5.min.css') }}>
<link rel="stylesheet" href="{{ asset('admin/compiled/css/table-datable-jquery.css') }}>

@endpush

@section('main')

<header class="py-5 min-vh-100">

    <div class="container">
        <div class="row">
            <div class="col-lg-8 mb-3">
                <div class="card mb-3">
                    <div class="card-body">
                        <h5 class="text-dark mb-3">Data @yield('title')</h5>
                        <hr>
                        <div class="table-responsive">
                            <table class="table table-striped table-bordered w-100"
id="cart-table">
                                <thead>
                                    <tr class="text-center bg-esa-secondary text-white">
                                        <th width="5%">#</th>
                                        <th width="30%">Barang</th>
                                        <th width="15%">Harga</th>
                                        <th width="15%">Quantity</th>
                                        <th width="15%">Jumlah</th>
                                        <th width="15%">Aksi</th>
                                </tr>

```

```

        </tr>

        </thead>

        <tbody>

        </tbody>

        </table>

        </div>

        </div>

        </div>

        </div>

<div class="col-lg-4 mb-3">
    <div class="card">
        <div class="card-body">
            <h5 class="text-dark mb-3">Total</h5>
            <hr>
            <table class="table table-striped table-bordered">
                <tbody>
                    <tr>
                        <td>Total Harga</td>
                        <td class="text-end" id="total-harga">{{formatRupiah($cart->sum('jumlah')) }}</td>
                    </tr>
                    <tr>
                        <th>Total Bayar</th>
                        <th class="text-end" id="total-bayar">{{formatRupiah($cart->sum('jumlah')) }}</th>
                    </tr>
                </tbody>
            </table>
        </div>
    </div>
</div>

```

```
@push('scripts')  
  
    <script src="{!! asset('admin/extensions/sweetalert2/sweetalert2.min.js') !!}"></script>  
  
    <script src="{!! asset('admin/extensions/datatables.net/js/jquery.dataTables.min.js') !!}"></script>
```

```
<script src="{!! asset('admin/extensions/datatables.net-bs5/js/dataTables.bootstrap5.min.js') !!}"></script>

<script src="https://app.sandbox.midtrans.com/snap/snap.js"
       data-client-key="{!! config('services.midtrans.clientKey') !!}"></script>

<script>
$(document).ready(function() {

    datatableCall('cart-table', '{!! route('cart.index') !!}', [
        {
            data: 'DT_RowIndex',
            name: 'DT_RowIndex'
        },
        {
            data: 'barang',
            name: 'barang'
        },
        {
            data: 'harga',
            name: 'harga'
        },
        {
            data: 'quantity',
            name: 'quantity'
        },
        {
            data: 'jumlah',
            name: 'jumlah'
        }
    ]);
});
```

```
        },
        {
            data: 'aksi',
            name: 'aksi'
        }
    ]);
}

$("#saveData").submit(function(e) {
    setButtonLoadingState("#saveData .btn.btn-primary", true);
    e.preventDefault();
    const kode = $("#saveData #id").val();
    const url = `/cart/${kode}`;
    const data = new FormData(this);
    data.append("_method", "PUT");
    const successCallback = function(response) {
        getTotalCart();
        setButtonLoadingState("#saveData .btn.btn-primary", false);
        handleSuccess(response, "cart-table", "createModal");
    };
    const errorCallback = function(error) {
        setButtonLoadingState("#saveData .btn.btn-primary", false);
        handleValidationErrors(error, "saveData", ["quantity"]);
    };
});
```

```
};

ajaxCall(url, "POST", data, successCallback, errorCallback);

});

$("#payMidtrans").submit(function(e) {
    e.preventDefault();
    Swal.fire({
        title: 'Konfirmasi Pembayaran',
        text: "Apakah Anda yakin akan melanjutkan transaksi pembayaran?",
        icon: 'warning',
        showCancelButton: true,
        confirmButtonColor: '#3085d6',
        cancelButtonColor: '#d33',
        confirmButtonText: 'Ya, Lanjutkan!',
        cancelButtonText: 'Batal'
    }).then((result) => {
        if (result.isConfirmed) {
            processPayment();
        }
    });
});

function processPayment() {
```

```
setButtonLoadingState("#payMidtrans .btn.btn-success", true,  
"Bayar");  
  
const url = "{{ route('transaksi.index') }}";  
  
const successCallback = function(response) {  
    setButtonLoadingState("#payMidtrans .btn.btn-success", false,  
"Bayar");  
  
    snap.pay(response.data.snapToken, {  
        onSuccess: function(result) {  
            handleSuccess("Transaksi Berhasil", null, null, "/transaksi");  
        },  
        onPending: function(result) {  
            handleSuccess("Transaksi Berhasil", null, null, "/transaksi");  
        },  
        onError: function(result) {  
            console.log(result)  
        }  
    });  
};  
  
const errorCallback = function(error) {  
    setButtonLoadingState("#payMidtrans .btn.btn-success", false,  
"Bayar");  
    handleValidationErrors(error, "payMidtrans");
```

```

    };

    ajaxCall(url, "POST", null, successCallback, errorCallback);

}

});

</script>

@endpush

```

4. Script pada tampilan modal yang mana disini berfungsi sebagai form tambah dan edit pada user.edit data quantity pada keranjang

```

<div class="modal fade" id="createModal" tabindex="-1" aria-
labelledby="exampleModalLabel" aria-hidden="true"
data-bs-backdrop="static" data-bs-keyboard="false">
<div class="modal-dialog modal-dialog-centered">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title" id="exampleModalLabel"><span id="label-
modal"></span> Data @yield('title')
</h5>
<button type="button" class="btn-close" data-bs-dismiss="modal" aria-
label="Close"></button>
</div>
<form id="saveData" autocomplete="off">
<div class="modal-body">
<input type="hidden" id="id">

```

```

<div class="form-group mb-3">
    <label for="quantity" class="form-label">Quantity <span
        class="text-danger">*</span></label>
    <input type="number" class="form-control" id="quantity"
        name="quantity">
    <small class="invalid-feedback" id="errorquantity"></small>
</div>
</div>

<div class="modal-footer">
    <button type="button" class="btn btn-secondary" data-bs-
        dismiss="modal">Batal</button>
    <button type="submit" class="btn btn-primary">Simpan</button>
</div>
</form>
</div>
</div>
</div>

```



5. Script pada proses edit quantity disini ada beberapa validasi untuk data yang diinput untuk dapat mendapatkan data yang valid dan disimpan pada database.

```

public function store(Request $request)
{
    $validator = Validator::make($request->all(), [
        'barang' => 'required|exists:barangs,id',

```

```
]);
```

```
if ($validator->fails()) {  
    return $this->errorResponse($validator->errors(), 'Data tidak valid.', 422);  
}
```

```
$barang = Barang::find($request->barang);  
  
$cart = DetailTransaksi::where('barang_id', $request->barang)->whereNull('transaksi_id')->where('cart_id', auth()->user()->id)->first();
```

```
if (!$cart) {
```

```
$cart = DetailTransaksi::create([  
    'cart_id' => auth()->user()->id,  
    'barang_id' => $request->barang,  
    'quantity' => 1,  
    'jumlah' => 1 * $barang->harga,  
]);
```

```
} else {
```

```
$quantity = $cart->quantity + 1;  
  
$cart->update([  
    'quantity' => $quantity,  
    'jumlah' => $quantity * $barang->harga,  
]);
```

```
}
```

```
$count = DetailTransaksi::whereNull('transaksi_id')->where('cart_id', auth()->user()->id)->count();  
  
return $this->successResponse(compact('cart', 'count'), 'Data ditambahkan  
ke keranjang.');
```

}



2.5 Rancangan Laporan

A. Membuat Laporan Transaksi Admin Per Periode

Langkah – langkah untuk membuat laporan transaksi per periode adalah sebagai berikut :

1. Membuat script untuk mengeksekusi laporan transaksi periode yang ada pada TransaksiController dengan function index yang menampung beberapa request yang mana disini mengambil data bulan dan tahun. Melakukan query dengan model berdasarkan data transaksi user.

```
public function index(Request $request)
{
    $bulan = $request->bulan ?? date('m');
    $tahun = $request->tahun ?? date('Y');

    $transaksis = Transaksi::with('user')->whereMonth('created_at', $bulan)-
>whereYear('created_at', $tahun)->latest()->get();

    if ($request->ajax()) {
        if ($request->mode == "datatable") {

            return DataTables::of($transaksis)
                ->addColumn('total', function ($transaksi) {
                    return formatRupiah($transaksi->total);
                })
                ->addColumn('tanggal', function ($transaksi) {
                    return formatTanggal($transaksi->created_at);
                })
        }
    }
}
```

```

->addColumn('customer', function ($transaksi) {
    return $transaksi->user->nama;
})

->addColumn('status', function ($transaksi) {
    return formatStatusLabel($transaksi->status);
})

->addColumn('aksi', function ($transaksi) {
    $detailButton = '<a class="btn btn-sm btn-info me-1 d-inline-flex"
href="/admin/transaksi/' . $transaksi->id . '"><i class="bi bi-info-circle me-1"></i>Detail</a>';

    if ($transaksi->status == 'pending') {
        $confirmationButton = '<button class="btn btn-sm btn-success d-
inline-flex" onclick="confirmTransaction(' . $transaksi->id . ')"><i class="bi bi-
check-circle me-1"></i>Konfirmasi</button>';
    }

    return $detailButton . $confirmationButton;
} elseif ($transaksi->status == 'disetujui') {
    $cetakButton = '<a class="btn btn-sm btn-primary me-1 d-inline-
flex" href="/admin/transaksi/struk/' . $transaksi->id . '"><i class="bi bi-printer me-1"></i>Cetak</a>';

    return $detailButton . $cetakButton;
} else {
    return $detailButton;
}
})

->addIndexColumn()

```

```

->rawColumns(['total', 'tanggal', 'customer', 'status', 'aksi'])

->make(true);

}

}

if ($request->mode == "pdf") {

$bulanTahun = formatTanggal($tahun . "-" . $bulan . "-01", 'F Y');

$pdf = PDF::loadView('pages.admin.transaksi.pdf', ['transaksis' =>

$transaksis->where('status', 'disetujui'), 'bulanTahun' => $bulanTahun]);

$options = [
    'margin_top' => 0,
    'margin_right' => 0,
    'margin_bottom' => 0,
    'margin_left' => 0,
];

$pdf->setOptions($options);

$pdf->setPaper('a4', 'landscape');

return $pdf->stream('Laporan Transaksi ' . $bulanTahun . '.pdf');

}

return view('pages.admin.transaksi.index');

}

```

2. Menambahkan tampilan untuk memilih bulan dan tahun dan tombol laporan pada admin presensi untuk laporan transaksi per periode.

```
@extends('layouts.admin')
```

```
@section('title', 'Transaksi')
```

```
@push('style')
```

```
<link rel="stylesheet" href="{{ asset('admin/extensions/datatables.net-bs5/css/dataTables.bootstrap5.min.css') }}">
```

```
<link rel="stylesheet" href="{{ asset('admin/compiled/css/table-datatable-jquery.css') }}">
```

```
<link rel="stylesheet" href="{{ asset('admin/extensions/sweetalert2/sweetalert2.min.css') }}">
```

```
@endpush
```

```
@section('main')
```

```
<div class="content-wrapper container">
```

```
    <div class="page-heading">
```

```
        <div class="row">
```

```
            <div class="col-12 col-md-6 order-md-1 order-last">
```

```
                <h3>@yield('title')</h3>
```

```
            </div>
```

```
            <div class="col-12 col-md-6 order-md-2 order-first">
```

```

<nav aria-label="breadcrumb" class="breadcrumb-header float-start
float-lg-end">

    <ol class="breadcrumb">
        <li class="breadcrumb-item"><a
            href="/admin">Dashboard</a></li>
        <li class="breadcrumb-item active" aria-
current="page">@yield('title')</li>
    </ol>
</nav>

</div>
</div>
</div>
</div>
</div>
</div>
<div class="page-content">
    <section class="row">
        <div class="col-12">
            <div class="card">
                <div class="card-body">
                    <div class="d-flex justify-content-between align-items-center">
                        <h5 class="card-title mb-0">Data @yield('title')</h5>
                        <div>
                            <a id="downloadPdf" class="btn btn-sm btn-danger"
                                target="_blank"><i
                                class="bi bi-file-pdf me-2"></i>Laporan</a>
                        </div>

```

```

</div>

<hr>

<div class="row mb-3">

<div class="col-lg-6">

<div class="form-group">

<label for="bulan_filter" class="form-label">Bulan</label>

<select name="bulan_filter" id="bulan_filter" class="form-control">

@foreach (bulan() as $key => $value)

    <option value="{{ $key + 1 }}"{{ $key + 1 == date('m') ? 'selected' : '' }}>
        {{ $value }}</option>

@endforeach

</select>

</div>

</div>

<div class="col-lg-6">

<div class="form-group">

<label for="tahun_filter" class="form-label">Tahun</label>

<select name="tahun_filter" id="tahun_filter" class="form-control">

@for ($i = now()->year; $i >= now()->year - 4; $i--)

    <option value="{{ $i }}"{{ $i == date('Y') ? 'selected' : '' }}>
        {{ $i }}</option>

@endfor

</select>

</div>

```

```

        </option>

    @endfor

    </select>

</div>

</div>

<div class="table-responsive">

    <table class="table table-bordered table-striped" id="transaksi->
table" width="100%">

        <thead>
            <tr>
                <th width="5%">#</th>
                <th>Tanggal</th>
                <th>Kode Transaksi</th>
                <th>Customer</th>
                <th>Status</th>
                <th>Total</th>
                <th width="15%">Aksi</th>
            </tr>
        </thead>
        <tbody>
            </tbody>
    </table>
</div>
</div>

```

```

        </div>

        </div>

    </section>

</div>

</div>

@endsection

@push('scripts')

<script src="{!! asset('admin/extensions/datatables.net/js/jquery.dataTables.min.js') !!}"></script>

<script src="{!! asset('admin/extensions/datatables.net-bs5/js/dataTables.bootstrap5.min.js') !!}"></script>

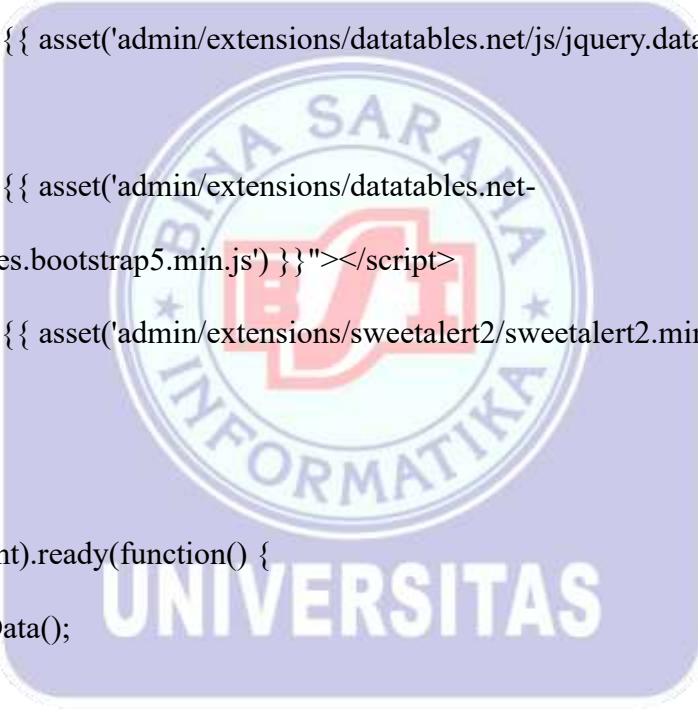
<script src="{!! asset('admin/extensions/sweetalert2/sweetalert2.min.js') !!}"></script>

<script>

$(document).ready(function() {
    renderData();
}

datatableCall('transaksi-table', '{ route('admin.transaksi.index') }', [
    {
        data: 'DT_RowIndex',
        name: 'DT_RowIndex'
    },
    {
        data: 'tanggal',
        name: 'tanggal'
    }
]);

```



```
        },
        {
            data: 'kode_transaksi',
            name: 'kode_transaksi'
        },
        {
            data: 'customer',
            name: 'customer'
        },
        {
            data: 'status',
            name: 'status'
        },
        {
            data: 'total',
            name: 'total'
        },
        {
            data: 'aksi',
            name: 'aksi'
        },
    ],
);

$("#bulan_filter, #tahun_filter").on("change", function() {
    renderData();
})
```



```
});  
});  
  
const renderData = () => {  
    const downloadPdf =  
  
        `/admin/transaksi?mode=pdf&bulan=${$("#bulan_filter").val()}&tahun=${$("#tahu  
n_filter").val()}`;  
  
    $("#downloadPdf").attr("href", downloadPdf);  
}  
  
const confirmTransaction = (id) => {  
    Swal.fire({  
        title: 'Konfirmasi Transaksi',  
        text: 'Apakah anda yakin ingin mengkonfirmasi transaksi ini?',  
        icon: 'warning',  
        showCancelButton: true,  
        confirmButtonColor: '#3085d6',  
        cancelButtonColor: '#d33',  
        confirmButtonText: 'Setujui',  
        showDenyButton: true,  
        denyButtonColor: '#ffc107',  
        denyButtonText: 'Batalkan',  
        showCloseButton: true,  
        closeButtonAriaLabel: 'Batalkan'  
    })  
};
```

```
}).then((result) => {

    const kode = id;

    const url = `/admin/transaksi/${kode}`;

    const data = new FormData();

    if (result.isConfirmed) {

        data.append("_method", "PUT");

        data.append("status", "disetujui");

        const successCallback = function(response) {

            handleSuccess(response, "transaksi-table");

        };

        const errorCallback = function(error) {

            handleValidationErrors(error, "transaksi-table");

        };

        ajaxCall(url, "POST", data, successCallback, errorCallback);

    } else if (result.isDenied) {

        data.append("_method", "PUT");

        data.append("status", "ditolak");

        const successCallback = function(response) {

            handleSuccess(response, "transaksi-table");

        };

    }

}
```

```

        const errorCallback = function(error) {
            handleValidationErrors(error, "transaksi-table");
        };

        ajaxCall(url, "POST", data, successCallback, errorCallback);
    } else {
        Swal.close();
    }
});

}

</script>
@endpush

```

3. Membuat tampilan cetak pdf pada laporan transaksi per periode dengan menampilkan user.

```
@extends('layouts.pdf')
```

```
@section('title', 'Laporan Transaksi ' . $bulanTahun)
```

```
@push('style')
```

```
@endpush
```

```
@section('main')
```

```
<div>
```

```

<table width="100%" border="1" cellpadding="2.5" cellspacing="0">

<thead>

<tr>

<th width="5%">#</th>

<th>Tanggal</th>

<th>Kode Transaksi</th>

<th>Customer</th>

<th>Status</th>

<th>Total</th>

</tr>

</thead>

<tbody valign="top">

@foreach ($transaksis as $transaksi)

<tr>

<td align ="center">{{ $loop->iteration }}</td>

<td>{{ formatTanggal($transaksi->created_at) }}</td>

<td>{{ $transaksi->kode_transaksi }}</td>

<td>{{ $transaksi->user->nama }}</td>

<td>{{ $transaksi->status }}</td>

<td>{{ formatRupiah($transaksi->total) }}</td>

</tr>

@endforeach

</tbody>

<tfoot>

<tr>

```

```

<th colspan="5">Total</th>
<th>{{ formatRupiah($transaksis->sum('total')) }}</th>
</tr>
</tfoot>
</table>
</div>
@endsection

```

@push('scripts')

@endpush

4. Melakukan pengetesan laporan dengan mengakses halaman admin transaksi dan memilih data bulan Juli dengan tahun 2024



Gambar II. 46
Laporan Transaksi

- Klik tombol “Laporan” yang mana akan mengarahkan ke halaman laporan transaksi per periode sesuai dengan data yang dipilih.

Laporan Transaksi Juli 2024

#	Tanggal	Kode Transaksi	Customer	Status	Total
1	Jumat, 19 Juli 2024	ESA-240719-9628812	rina	diselesaikan	Rp 900.000
2	Jumat, 19 Juli 2024	ESA-240719-1771187	wida	diselesaikan	Rp 300.000
3	Jumat, 19 Juli 2024	ESA-240719-9013811	wida	dipesan	Rp 100.000.000
4	Jumat, 19 Juli 2024	ESA-240719-3911340	rina	diselesaikan	Rp 250.000.000
5	Jumat, 19 Juli 2024	ESA-240719-0888006	rina	dipesan	Rp 110.000.000
6	Jumat, 19 Juli 2024	ESA-240719-2116788	wida	dipesan	Rp 180.000.000
7	Rabu, 17 Juli 2024	ESA-240717-9670938	Administrator	diselesaikan	Rp 100.000.000
Total					Rp 481.400.000

Gambar II. 47
Laporan Transaksi

B. Membuat Laporan Transaksi User

Langkah – langkah untuk membuat laporan transaksi per periode adalah sebagai berikut :

1. Membuat script untuk mengeksekusi laporan transaksi periode yang ada pada TransaksiController dengan function index yang menampung beberapa request yang mana disini mengambil data bulan dan tahun. Melakukan query dengan model berdasarkan data transaksi userMenambahkan tampilan untuk memilih bulan dan tahun dan tombol laporan pada admin izin untuk laporan izin per periode

```
public function index(Request $request)
{
    $bulan = $request->bulan ?? date('m');
    $tahun = $request->tahun ?? date('Y');

    $transaksis = Transaksi::with('user')->whereMonth('created_at', $bulan)->whereYear('created_at', $tahun)->where('user_id', auth()->user()->id)->latest()->get();

    if ($request->ajax()) {
        if ($request->mode == "datatable") {

            return DataTables::of($transaksis)
                ->addColumn('total', function ($transaksi) {
                    return formatRupiah($transaksi->total);
                })
                ->addColumn('tanggal', function ($transaksi) {
                    return formatTanggal($transaksi->created_at);
                })
        }
    }
}
```

```

->addColumn('status', function ($transaksi) {
    return formatStatusLabel($transaksi->status);
})

->addColumn('aksi', function ($transaksi) {
    $detailButton = '<a class="btn btn-sm btn-info me-1 d-inline-flex" href="/transaksi/' . $transaksi->id . '"><i class="bi bi-info-circle me-1"></i>Detail</a>';

    if ($transaksi->status == 'disetujui') {
        $cetakButton = '<a class="btn btn-sm btn-primary me-1 d-inline-flex" href="/transaksi/struk/' . $transaksi->id . '"><i class="bi bi-printer me-1"></i>Cetak</a>';
        return $detailButton . $cetakButton;
    } else {
        return $detailButton;
    }
})

->addIndexColumn()
->rawColumns(['total', 'tanggal', 'status', 'aksi'])

->make(true);

}

if ($request->mode == "pdf") {
    $bulanTahun = formatTanggal($tahun . "-" . $bulan . "-01", 'F Y');
}

```

```

$pdf = PDF::loadView('pages.admin.transaksi.pdf', ['transaksis' =>
$transaksis->where('status', 'disetujui'), 'bulanTahun' => $bulanTahun]);

$options = [
    'margin_top' => 0,
    'margin_right' => 0,
    'margin_bottom' => 0,
    'margin_left' => 0,
];

$pdf->setOptions($options);
$pdf->setPaper('a4', 'landscape');

return $pdf->stream('Laporan Transaksi ' . $bulanTahun . '.pdf');
}

return view('pages.frontend.transaksi.index');
}

```



2. Menambahkan tampilan untuk memilih bulan dan tahun dan tombol cetak pada user untuk laporan per transaksi

```
@extends('layouts.frontend')
```

```
@section('title', 'Transaksi')
```

```

@push('style')

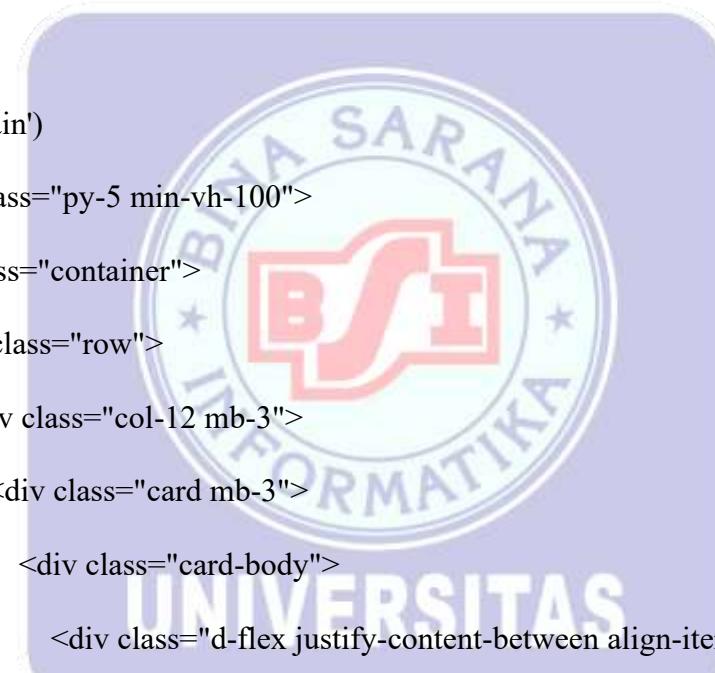
<link rel="stylesheet" href="{{ asset('admin/extensions/sweetalert2/sweetalert2.min.css') }}>

<link rel="stylesheet" href="{{ asset('admin/extensions/datatables.net-bs5/css/dataTables.bootstrap5.min.css') }}>

<link rel="stylesheet" href="{{ asset('admin/compiled/css/table-datable-jquery.css') }}>

@endpush

```



```

@section('main')

<header class="py-5 min-vh-100">

<div class="container">

<div class="row">

<div class="col-12 mb-3">

<div class="card mb-3">

<div class="card-body">

<div class="d-flex justify-content-between align-items-center">

<h5 class="card-title mb-0">Data @yield('title')</h5>

<div>

<a id="downloadPdf" class="btn btn-sm btn-danger" target="_blank"><i

class="bi bi-file-pdf me-2"></i>Laporan</a>

</div>

</div>

<hr>

```

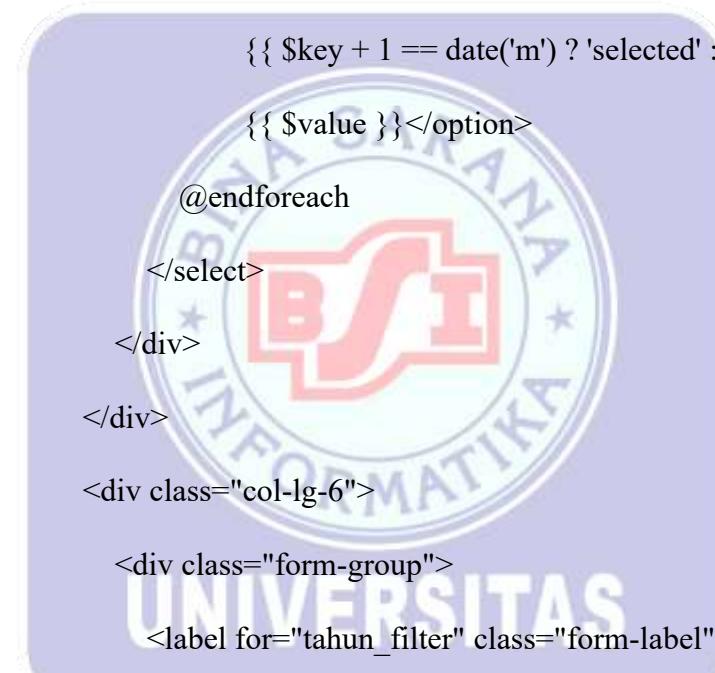
```

<div class="row mb-3">

    <div class="col-lg-6">
        <div class="form-group">
            <label for="bulan_filter" class="form-label">Bulan</label>
            <select name="bulan_filter" id="bulan_filter" class="form-control">
                @foreach (bulan() as $key => $value)
                    <option value="{{ $key + 1 }}" {{ $key + 1 == date('m') ? 'selected' : '' }}>
                        {{ $value }}</option>
                @endforeach
            </select>
        </div>
    </div>

    <div class="col-lg-6">
        <div class="form-group">
            <label for="tahun_filter" class="form-label">Tahun</label>
            <select name="tahun_filter" id="tahun_filter" class="form-control">
                @for ($i = now()->year; $i >= now()->year - 4; $i--)
                    <option value="{{ $i }}" {{ $i == date('Y') ? 'selected' : '' }}>
                        {{ $i }}</option>
                @endfor
            </select>
        </div>
    </div>

```




```

@push('scripts')

<script src="{{ asset('admin/extensions/sweetalert2/sweetalert2.min.js') }}></script>

<script src="{{ asset('admin/extensions/datatables.net/js/jquery.dataTables.min.js') }}></script>

<script src="{{ asset('admin/extensions/datatables.net-bs5/js/dataTables.bootstrap5.min.js') }}></script>

<script>

$(document).ready(function() {
    renderData();
    datatableCall('transaksi-table', '{{ route('transaksi.index') }}', [
        {
            data: 'DT_RowIndex',
            name: 'DT_RowIndex'
        },
        {
            data: 'tanggal',
            name: 'tanggal'
        },
        {
            data: 'kode_transaksi',
            name: 'kode_transaksi'
        },
        {
    ]
});

```

```

        data: 'status',
        name: 'status'
    },
    {
        data: 'total',
        name: 'total'
    },
    {
        data: 'aksi',
        name: 'aksi'
    }
]);
$("#bulan_filter, #tahun_filter").on("change", function() {
    renderData();
});
});
```



```

const renderData = () => {
    const downloadPdf =
        `/transaksi?mode=pdf&bulan=${$("#bulan_filter").val()}&tahun=${$("#tahun_filter").val()}`;
    $("#downloadPdf").attr("href", downloadPdf);
}
```

```
</script>
```

```
@endpush
```

3. Membuat tampilan cetak pdf pada laporan transaksi user per transaksi dengan menampilkan data transaksi user yang disetujui.

```
@extends('layouts.pdf')
```

```
@section('title', 'Laporan Transaksi ' . $bulanTahun)
```

```
@push('style')
```

```
@endpush
```

```
@section('main')
```

```
<div>
```

```
<table width="100%" border="1" cellpadding="2.5" cellspacing="0">
```

```
<thead>
```

```
<tr>
```

```
<th width="5%">#</th>
```

```
<th>Tanggal</th>
```

```
<th>Kode Transaksi</th>
```

```
<th>Customer</th>
```

```
<th>Status</th>
```

```
<th>Total</th>
```

```
</tr>
```

```
</thead>
```



```

<tbody valign="top">

    @foreach ($transaksis as $transaksi)

        <tr>

            <td align ="center">{{ $loop->iteration }}</td>

            <td>{{ formatTanggal($transaksi->created_at) }}</td>

            <td>{{ $transaksi->kode_transaksi }}</td>

            <td>{{ $transaksi->user->nama }}</td>

            <td>{{ $transaksi->status }}</td>

            <td>{{ formatRupiah($transaksi->total) }}</td>

        </tr>

    @endforeach

</tbody>

<tfoot>

    <tr>

        <th colspan="5">Total</th>

        <th>{{ formatRupiah($transaksis->sum('total')) }}</th>

    </tr>

</tfoot>

</table>

</div>

@endsection

@push('scripts')

@endpush

```

4. Melakukan pengetesan laporan dengan mengakses halaman admin transaksi dan memilih data bulan Juli dengan tahun 2024.

Laporan Transaksi Juli 2024

#	Tanggal	Kode Transaksi	Customer	Status	Total
1	Minggu, 21 Juli 2024	EKA-240721-9673756	widaaryanti	diselesaikan	Rp 210.000.000
Total					Rp 210.000.000

Gambar II. 48

Laporan Transaksi



2.6 Pengujian Unit

Pengujian terhadap program yang mana disini terdapat beberapa validasi menggunakan metode blackbox testing yang berfokus terhadap data yang dimasukan. Menurut (Perdana et al., 2024) Pengujian black box bertujuan untuk menguji aplikasi secara menyeluruh, memeriksa fitur, usecase, dan fungsionalitasnya. Karena berfokus pada perilaku eksternal perangkat lunak tanpa memerhatikan struktur internalnya, black box testing adalah teknik pengujian yang cukup mudah digunakan.

A. Pengujian Terhadap Form Login

Tabel II. 35

Pengujian Unit Form Login

No	Skenario Pengujian	Test case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Email dan Password tidak diisi kemudian klik tombol Login	Email:(kosong) Password:(kosong)	Sistem otomatis menolak data login dengan tampilan pesan validasi form “Email harus diisi.” dan “Password harus diisi”	Sesuai Harapan	Valid

2	Email diisi terdaftar di tabel users dan Password tidak diisi kemudian klik tombol Login	Email: wida@gmail.com Password: (kosong)	Sistem akan menolak data login dengan menampilkan pesan validasi form “Password harus diisi”	Sesuai Harapan	Valid
3	Email tidak diisi dan Password diisi kemudian klik tombol Login	Email: (kosong) Password: 11221122	Sistem akan menolak data login dengan menampilkan pesan validasi form “Email harus diisi”	Sesuai Harapan	Valid
4	Email diisi namun tidak terdaftar di tabel users	Email: wid@gmail.com Password : 112211	Sistem akan menolak data login dengan menampilkan pesan	Sesuai Harapan	Valid

	dan Password diisi kurang dari 8 digit kemudian klik tombol Login		validasi form “Pilihan email yang dipilih tidak valid.” dan “Password harus setidaknya 8 karakter.”		
5	Email diisi terdaftar di tabel users dan Password diisi yang tidak sesuai dengan di tabel user kemudian klik tombol Login	Email: wida@gmail.com Password: 12345678	Sistem akan menolak data login dengan menampilkan pesan “Password Tidak Valid”	Sesuai Harapan	Valid

6	Email diisi dan Password diisi yang terdaftar di tabel user kemudian klik tombol Login	Email: wida@gmail.com Password: 11221122	Sistem akan menerima data login dengan menampilkan pesan “Login berhasil”	Sesuai Harapan	Valid
---	--	---	---	----------------	-------

Sumber : Hasil Penelitian (2024)

B. Pengujian Terhadap Form Lupa Password

Tabel II. 36

Pengujian Unit Form Lupa Password

No	Skenario Pengujian	Test case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Email tidak diisi kemudian klik tombol Lupa Password	Email : kosong	Sistem akan menolak data lupa password dengan menampilkan pesan	Sesuai harapan	Valid

			validasi form “Email harus diisi.”		
2	Email diisi dengan email yang tidak terdaftar di tabel user kemudian klik tombol Lupa Password	Email : aryanti@gmail.com	Sistem akan menolak data lupa password dengan menampilkan pesan validasi form “Pilihan email yang dipilih tidak valid.”	Sesuai harapan	Valid
3	Email diisi dengan email yang terdaftar di tabel user kemudian klik tombol	Email : widaar@gmail.com	Sistem akan menerima data lupa password dengan menampilkan pesan “Reset password”	Sesuai harapan	Valid

	Lupa Password		email terkirim”		
--	------------------	--	--------------------	--	--

Sumber : Hasil Penelitian (2024)

C. Pengujian Terhadap Form Reset Password

Tabel II. 37

Pengujian Unit Form Reset Password

No	Skenario Pengujian	Test case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Password dan Konfirmasi Password tidak diisi kemudian klik tombol Reset Password	Password : (kosong) Konfirmasi Password : (kosong)	Sistem akan menolak data reset password dengan menampilkan pesan validasi form “Password harus diisi.” dan “Konfirmasi Password harus diisi”	Sesuai harapan	Valid

2	<p>Password dan Konfirmasi</p> <p>Password diisi namun tidak sama datanya kemudian klik tombol Reset</p> <p>Password</p>	<p>Password : 12345678</p> <p>Konfirmasi</p> <p>Password : 09876543</p>	<p>Sistem akan menolak data reset password dengan menampilkan pesan validasi form</p> <p>“Password konfirmasi dan password harus cocok.”</p>	Sesuai harapan	Valid
3	<p>Password dan Konfirmasi</p> <p>Password diisi namun tidak 8 karakter,</p> <p>kemudian klik tombol Reset</p> <p>Password</p>	<p>Password : 1234567</p> <p>Konfirmasi</p> <p>Password : 1234567</p>	<p>Sistem akan menolak data reset password dengan menampilkan pesan validasi form</p> <p>“Password dan Konfirmasi Password”</p>	Sesuai harapan	Valid

			harus setidaknya 8 karakter.”		
4	Password dan Konfirmasi Password diisi dengan data yang sama kemudian klik tombol Reset Password	Password : 12345678 Konfirmasi Password : 12345678	Sistem akan menerima data reset password dengan menampilkan pesan “Password berhasil direset”	Sesuai harapan	Valid

Sumber : Hasil Penelitian (2024)

D. Pengujian Terhadap Form User

Tabel II. 38

Pengujian Unit Form User

No	Skenario Pengujian	Test case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Nama, Email, Password, Konfirmasi	Nama: (kosong) Email: (kosong) Password: (kosong)	Sistem akan menolak data reset password	Sesuai harapan	Valid

	Password, Role, No Hp, Alamat tidak diisi kemudian klik tombol Simpan pada form User	Konfirmasi Password: (kosong) Role : (kosong) No Hp : (kosong) Alamat: (kosong)	dengan menampilkan pesan validasi form "Nama harus diisi.", "Email harus diisi.", "Password harus diisi.", "Konfirmasi Password harus diisi.", "Role harus diisi.", "No Hp harus diisi.", “Alamat harus diisi.”		
2	Nama, Password, Konfirmasi Password, Role, No Hp, Alamat	Nama: (kosong) Email: wida@gmail.com Password: (kosong)	Sistem akan menolak data user dengan menampilkan pesan validasi form "Nama	Sesuai harapan	Valid

	tidak diisi, dan email diisi dengan email yang sudah ada kemudian klik tombol Simpan pada form User	Konfirmasi Password: (kosong) Role : (kosong) No Hp : (kosong) Alamat: (kosong)	harus diisi.", "Email sudah ada.", "Password harus diisi.", "Konfirmasi Password harus diisi.", "Role harus diisi.", "No Hp harus diisi.", “Alamat harus diisi.”		
3	Nama, Email, Password, Konfirmasi Password, Role, No Hp, Alamat diisi kemudian klik tombol	Nama: widaa Email: widaa@gmail.com Password: 12345678 Konfirmasi Password: 12345678 Role : User No Hp : 09876543	Sistem akan menerima data user menampilkan pesan “Data user berhasil ditambahkan”\	Sesuai harapan	Valid

	Simpan pada form User	Alamat: Ciamis			
4	Pengguna klik tombol edit pada user	Tombol edit user	Sistem akan menampilkan modal form edit data user yang dipilih	Sesuai harapan	Valid
5	Pengguna klik tombol hapus pada user dan memilih Cancel	Tombol hapus user konfirmasi Cancel	Sistem tidak akan menghapus data user yang dipilih	Sesuai harapan	Valid
6	Pengguna klik tombol hapus pada user dan memilih OK	Tombol hapus user konfirmasi OK	Sistem akan menghapus data user yang dipilih	Sesuai harapan	Valid

Sumber : Hasil Penelitian (2024)

E. Pengujian terhadap Form Kategori

Tabel II. 39

Pengujian Unit Form Kategori

No	Skenario Pengujian	Test case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Nama tidak diisi kemudian klik tombol Simpan pada form kategori	Nama : (kosong)	Sistem akan menolak data kategori dengan menampilkan pesan validasi form “Nama harus diisi.”	Sesuai Harapan	Valid
2	Nama diisi dengan data yang sama dengan nama kategori yang lain kemudian klik tombol Simpan pada form kategori	Nama : Funiture	Sistem akan menolak data kategori dengan menampilkan pesan validasi form “Nama sudah ada.”	Sesuai Harapan	Valid

3	Nama diisi dengan data terbaru yang tidak ada di tabel kategoris kemudian klik tombol Simpan pada form kategori	Nama : Kitchen Set	Sistem akan menerima data kategori dengan menampilkan pesan “Data kategori berhasil ditambahkan”	Sesuai Harapan	Valid
4	Pengguna klik tombol edit pada kategori	Tombol edit kategori	Sistem akan menampilkan modal form edit data kategori yang dipilih	Sesuai Harapan	Valid
5	Pengguna klik tombol hapus pada kategori dan memilih Cancel	Tombol hapus kategori konfirmasi Cancel	Sistem tidak akan menghapus data kategori yang dipilih	Sesuai Harapan	Valid
6	Pengguna klik tombol hapus	Tombol hapus	Sistem akan menghapus	Sesuai Harapan	Valid

	pada kategori dan memilih YA	kategori konfirmasi YA	data kategori yang dipilih		
--	------------------------------	------------------------	----------------------------	--	--

Sumber : Hasil Penelitian (2024)

F. Pengujian Terhadap Form Barang

Tabel II. 40

Pengujian Unit Form Barang

No	Skenario Pengujian	Test case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Nama, Kategori, Harga dan Deskripsi tidak diisi. Kemudian klik tombol Simpan pada form barang	Nama: (kosong) Kategori: (kosong) Harga: (kosong) Deskripsi: (kosong)	Sistem akan menolak data barang dengan menampilkan pesan validasi form “Nama harus diisi.” , “Kategori harus diisi”, “Harga harus diisi”, “Deskripsi harus diisi”	Sesuai Harapan	Valid

2	<p>Nama diisi dengan nama barang yang sudah ada di tabel barang, Kategori, Harga, dan Deskripsi tidak diisi. Kemudian klik tombol Simpan pada form barang</p>	<p>Nama: Meja Kategori: (kosong) Harga: (kosong) Deskripsi: (kosong)</p>	<p>Sistem akan menolak data barang dengan menampilkan pesan validasi form “Nama sudah ada.” , “Kategori harus diisi”, “Harga harus diisi”, “Deskripsi harus diisi”</p>	<p>Sesuai Harapan</p>	<p>Valid</p>
3	<p>Nama, Kategori, Harga dan Deskripsi diisi. Kemudian klik tombol Simpan pada form barang</p>	<p>Nama: Kursi Kategori: Furniture Harga: Rp.120.000 Deskripsi: Kursi Kayu</p>	<p>Sistem akan menerima data barang dengan menampilkan pesan “Data barang berhasil ditambahkan”</p>	<p>Sesuai Harapan</p>	<p>Valid</p>

4	Pengguna klik tombol edit pada barang	Tombol edit barang	Sistem akan menampilkan modal form edit data barang yang dipilih	Sesuai Harapan	Valid
5	Pengguna klik tombol hapus pada barang dan memilih Cancel	Tombol hapus barang konfirmasi Cancel	Sistem tidak akan menghapus data barang yang dipilih	Sesuai Harapan	Valid
6	Pengguna klik tombol hapus pada barang dan memilih YA	Tombol hapus barang konfirmasi YA	Sistem akan menghapus data barang yang dipilih	Sesuai Harapan	Valid

Sumber : Hasil Penelitian (2024)

G. Pengujian Terhadap Form Barang Gambar

Tabel II. 41

Pengujian Unit Form Barang Gambar

No	Skenario Pengujian	Test case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Gambar tidak diisi	Gambar: (kosong)	Sistem akan menolak data barang dengan menampilkan pesan validasi form "Gambar harus diisi."	Sesuai Harapan	Valid
2	Gambar diisi dengan format .pdf kemudian klik tombol Simpan pada form barang	Gambar: Manualbook.pdf	Sistem akan menolak data barang dengan menampilkan pesan validasi form "Gambar harus berupa gambar."	Sesuai Harapan	Valid
3	Gambar diisi dengan format .png kemudian	Gambar: Picture1	Sistem akan menolak data barang dengan menampilkan	Sesuai Harapan	Valid

	klik tombol Simpan pada form barang		pesan validasi form “ Data Barang Gambar Berhasil Ditambahkan”		
4	Pengguna klik tombol edit pada data galeri barang	Tombol edit barang gambar	Sistem akan menampilkan modal form edit data barang yang dipilih	Sesuai Harapan	Valid
4	Pengguna klik tombol hapus pada data galeri barang dan memilih Cancel	Tombol hapus barang gambar konfirmasi Cancel	Sistem tidak akan menghapus data barang yang dipilih	Sesuai Harapan	Valid
5	Pengguna klik tombol hapus pada	Tombol hapus barang konfirmasi YA	Sistem akan menghapus data barang yang dipilih	Sesuai Harapan	Valid

	barang dan memilih YA				
--	-----------------------	--	--	--	--

Sumber : Hasil Penelitian (2024)

H. Pengujian Terhadap Form Galeri

Tabel II. 42

Pengujian Unit Form Galeri

No	Skenario Pengujian	Test case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Nama, Tanggal Mulai, Tanggal Selesai dan Deskripsi tidak diisi. Kemudian klik tombol Simpan pada form galeri	Nama: (kosong) Tanggal Mulai: (kosong) Tanggal Selesai: (kosong) Deskripsi: (kosong)	Sistem akan menolak data galeri dengan menampilkan pesan validasi form “Nama harus diisi.” , “Tanggal Selesai harus diisi.”, “Tanggal Selesai harus diisi.”, “Deskripsi harus diisi.”	Sesuai Harapan	Valid

2	<p>Nama diisi dengan nama galeri yang sudah ada di tabel galeris, Tanggal Mulai, Tanggal Selesai dan Deskripsi tidak diisi. Kemudian klik tombol Simpan pada form galeri</p>	<p>Nama: Kolam Tanggal Mulai: (kosong) Tanggal Selesai: (kosong) Deskripsi: (kosong)</p>	<p>Sistem akan menolak data galeri dengan menampilkan pesan validasi form “Nama sudah ada” , “Tanggal ”, “Selesai harus diisi.”, “Deskripsi harus diisi.”</p>	Sesuai Harapan	Valid
3	<p>Nama, Tanggal Mulai, Tanggal Selesai dan Deskripsi diisi. Kemudian klik tombol Simpan pada form barang</p>	<p>Nama: Kolam Pribadi Tanggal Mulai: 02-07-2024 Tanggal Selesai: 02-08-2024</p>	<p>Sistem akan menerima data galeri dengan menampilkan pesan “Data Galeri berhasil ditambahkan”</p>	Sesuai Harapan	Valid

		Deskripsi: Kolam Pribadi			
4	Pengguna klik tombol edit pada galeri	Tombol edit galeri	Sistem akan menampilkan modal form edit data galeri yang dipilih	Sesuai Harapan	Valid
5	Pengguna klik tombol hapus pada galeri dan memilih Cancel	Tombol hapus galeri konfirmasi Cancel	Sistem tidak akan menghapus data galeri yang dipilih	Sesuai Harapan	Valid
6	Pengguna klik tombol hapus pada galeri dan memilih YA	Tombol hapus galeri konfirmasi YA	Sistem akan menghapus data galeri yang dipilih	Sesuai Harapan	Valid

Sumber : Hasil Penelitian (2024)

I. Pengujian Terhadap Form Galeri Gambar

Tabel II. 43

Pengujian Unit Form Galeri Gambar

No	Skenario Pengujian	Test case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Gambar tidak diisi	Gambar: (kosong)	Sistem akan menolak data galeri gambar dengan menampilkan pesan validasi form "Gambar harus diisi."	Sesuai Harapan	Valid
2	Gambar diisi dengan format .pdf kemudian klik tombol Simpan pada form galeri gambar	Gambar: Manualbook.pdf	Sistem akan menolak data galeri gambar dengan menampilkan pesan validasi form "Gambar harus berupa gambar."	Sesuai Harapan	Valid
3	Gambar diisi dengan	Gambar: Picture1	Sistem akan menolak data	Sesuai Harapan	Valid

	format .png kemudian klik tombol Simpan pada form galeri gambar		galeri gambar dengan menampilkan pesan validasi form “ Data Galeri Gambar Berhasil Ditambahkan”		
4	Pengguna klik tombol edit pada data galeri galeri gambar	Tombol edit galeri gambar	Sistem akan menampilkan modal form edit data galeri gambar yang dipilih	Sesuai Harapan	Valid
4	Pengguna klik tombol hapus pada data galeri galeri gambar dan memilih Cancel	Tombol hapus galeri gambar konfirmasi Cancel	Sistem tidak akan menghapus data galeri gambar yang dipilih	Sesuai Harapan	Valid

5	Pengguna klik tombol hapus pada galeri gambar dan memilih YA	Tombol hapus galeri gambar konfirmasi YA	Sistem akan menghapus data galeri gambar yang dipilih	Sesuai Harapan	Valid
---	--	--	---	----------------	-------

Sumber : Hasil Penelitian (2024)

J. Pengujian Terhadap Form Transaksi

Tabel II. 44

Pengujian Unit Form Transaksi

No	Skenario Pengujian	Test case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Melakukan konfirmasi status pemesanan	Konfirmasi Transaksi : Cancel	Sistem akan menampilkan kembali data transaksi	Sesuai Harapan	Valid
2	Melakukan konfirmasi status pemesanan	Konfirmasi Transaksi : Batalkan	Sistem akan menampilkan “Status Transaksi Berhasil Diupdate”	Sesuai Harapan	Valid

3	Melakukan konfirmasi status pemesanan	Konfirmasi Transaksi : Setujui	Sistem akan menampilkan “Status Transaksi Berhasil Diupdate”	Sesuai Harapan	Valid
4	Melihat detail pertransaksi		Sistem akan menampilkan halaman detail data per transaksi	Sesuai Harapan	Valid
5	Melakukan cetak pdf pada laporan		Sistem akan menampilkan laporan transaksi yang disetujui	Sesuai Harapan	Valid

Sumber : Hasil Penelitian (2024)

K. Pengujian terhadap Form Ubah Pengaturan

Tabel II. 45
Pengujian Unit Form Ubah Pengaturan

No	Skenario Pengujian	Test case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Nama, Email, No HP, Alamat dan Deskripsi tidak diisi. Kemudian klik tombol Simpan pada form pengaturan	Nama: (kosong) Email:(kosong) No Hp: (kosong) Alamat: (kosong) Deskripsi: (kosong)	Sistem akan menolak data pengaturan dengan menampilkan pesan validasi form “Nama harus diisi.” , “Email harus diis-3/.”, “No HP harus diisi.”, “Alamat harus diisi.”, “Deskripsi harus diisi.”	Sesuai Harapan	Valid
2	Nama, Email, No	Nama: Esa Gallery Email:wida@gmail.co	Sistem akan menolak data	Sesuai Harapan	Valid

	HP, Alamat diisi dan Deskripsi tidak diisi. Kemudian klik tombol Simpan pada form pengaturan	No Hp: 098765432 Alamat: Jl. Nasional 3, Sukamanah 04/02, sindangkasih, ciamis Deskripsi: (kosong)	pengaturan dengan menampilkan pesan validasi form “Deskripsi harus diisi.”		
3	Nama, Email, No HP, Alamat dan Deskripsi diisi. Kemudian klik tombol Simpan pada form pengaturan	Nama: Esa Gallery Email:wida@gmail.co No Hp: 098765432 Alamat: Jl. Nasional 3, Sukamanah 04/02, sindangkasih, ciamis Deskripsi: Esa Gallery	Sistem akan menerima data pengaturan dengan menampilkan pesan “Data Pengaturan berhasil diubah”	Sesuai Harapan	Valid

Sumber : Hasil Penelitian (2024)

L. Pengujian Terhadap Form Ubah Profile

Tabel II. 46

Pengujian Unit Form Ubah Profile

No	Skenario Pengujian	Test case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Nama, Email, tidak diisi. Kemudian klik tombol Simpan pada form profile	Nama: (kosong) Email: (kosong)	Sistem akan menolak data profile dengan menampilkan pesan validasi form “Nama harus diisi.”, “Email harus diisi.”	Sesuai Harapan	Valid
2	Nama diisi, Email tidak diisi. Kemudian klik tombol Simpan pada form profile	Nama: Administrator Email: (kosong)	Sistem akan menolak data profile dengan menampilkan pesan validasi form	Sesuai Harapan	Valid

			“Email harus diisi.”		
3	Nama, Email diisi. Kemudian klik tombol Simpan pada form profile	Nama: Administrator Email: widaar@gmail.com	Sistem akan menerima data pengaturan dengan menampilkan pesan “Data Profile berhasil diubah”	Sesuai Harapan	Valid
4	Password Lama, Password Baru dan Konfirmasi Password tidak diisi. Kemudian klik tombol Simpan pada form profile	Password Lama: (kosong) Password Baru: (kosong)	Sistem akan menolak data profile dengan menampilkan pesan validasi form “Password Lama harus diisi.”, “Password	Sesuai Harapan	Valid

			Lama harus diisi.”		
5	Password Lama, Password Baru diisi dan Konfirmasi Password tidak diisi. Kemudian klik tombol Simpan pada form profile	Nama: Administrator Email: wida@gmail.com Password Lama: 11221122 Password Baru: 12345678 Konfirmasi Password: (kosong)	Sistem akan menolak data profile dengan menampilkan pesan validasi form “Konfirmasi password tidak cocok”	Sesuai Harapan	Valid
6	Password Lama, Password Baru dan Konfirmasi Password diisi. Kemudian klik tombol	Nama: Administrator Email: wida@gmail.com Password Lama: 11221122 Password Baru: 12345678	Sistem akan menerima data pengaturan dengan menampilkan pesan “Data Password	Sesuai Harapan	Valid

	Simpan pada form profile	Konfirmasi Password: 12345678	berhasil diubah”		
--	--------------------------------	-------------------------------------	---------------------	--	--

Sumber : Hasil Penelitian (2024)



BAB III

KESIMPULAN

3.1 Kesimpulan

Esa Gallery adalah perusahaan yang bergerak di bidang jasa konstruksi, desain interior dan eksterior. Berdasarkan hasil riset, ditemukan bahwa sistem manajemen proyek mereka saat ini masih menggunakan metode manual. Hal ini menyebabkan berbagai masalah seperti penyimpanan dokumen proyek dan desain yang tersebar dan tidak terpusat, serta media visual yang tidak terstruktur dengan baik. Kondisi ini meningkatkan risiko kehilangan atau kerusakan data dan menghambat efisiensi operasional perusahaan.

Untuk mengatasi masalah ini, diusulkan pengembangan aplikasi sistem informasi bernama Explora. Aplikasi ini dirancang untuk menyimpan semua dokumen dan media secara terpusat dan terorganisir, serta meningkatkan efisiensi dan keamanan dalam pengelolaan proyek. Pendekatan yang digunakan adalah metode manajemen proyek berbasis Scrum, yang fleksibel dan adaptif terhadap perubahan, serta teknik observasi partisipatif untuk mendapatkan wawasan yang lebih akurat tentang kebutuhan pengguna dan tantangan yang dihadapi.

Penelitian sebelumnya menunjukkan adanya kekurangan dalam manajemen media visual dan dokumen dalam konteks seni dan pameran, serta kurangnya pengarsipan digital yang terstruktur. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan sistem manajemen proyek yang lebih efisien dan terstruktur, dengan fokus pada manajemen media visual dan dokumen. Aplikasi Explora diharapkan dapat meminimalkan risiko kehilangan data dan memastikan aksesibilitas.

Pengembangan sistem informasi ini diharapkan dapat membantu pemasaran dan meningkatkan daya saing Esa Gallery. Sistem ini harus stabil, cepat, aman, dan mampu

mendukung tujuan perusahaan dalam memasarkan produk, meningkatkan keunggulan kompetitif, dan berkontribusi pada pertumbuhan bisnis. Dengan demikian, implementasi aplikasi Explora akan memperbaiki masalah pengelolaan proyek saat ini, meningkatkan efisiensi operasional, dan mendukung pencapaian tujuan bisnis Esa Gallery di industri jasa konstruksi dan desain.

3.2 Saran

Untuk meningkatkan efisiensi dan aktivitas EXPLORA, disarankan untuk mengembangkan sistem dengan menambahkan fitur pengembalian uang atau pengembalian saldo melalui platform e-commerce Esa Gallery. Ini adalah fitur yang akan memungkinkan pelanggan untuk mendapatkan kembali dana mereka jika pesanan mereka dibatalkan atau produk dikembalikan. Dalam praktiknya, transaksi harus melibatkan pihak ketiga yang bertindak sebagai escrow, yang menjamin dana pelanggan aman sampai transaksi selesai dan diterima dengan baik. Jika ada masalah, dana tersebut dapat dikembalikan ke saldo akun pelanggan atau rekening bank mereka sesuai kebijakan yang berlaku. Fitur ini meningkatkan transparansi dan keamanan transaksi dan meningkatkan kepercayaan pelanggan terhadap layanan Esa Gallery.

DAFTAR PUSTAKA

- Batara, B. D., Sudiatmika, I. P. G. A., & Januhari, N. N. U. (2024). Sistem Informasi Pendataan Barang Pada Toko Adi Ayu Berbasis Web. *Seminar Hasil Penelitian Informatika Dan Komputer (SPINTER)| Institut Teknologi Dan Bisnis STIKOM Bali*, 442–447.
- Cv, D. I., & Jaya, M. (2024). *SISTEM INFORMASI PENYEWAAN TRUK BERBASIS WEB*. 10(1), 57–63.
- David, N., Muntahanah, M., & Darmi, Y. (2024). RANCANG BANGUN APLIKASI SELEKSI PESERTA OLIMPIADE SAINS MENGGUNAKAN ALGORITMA PROMETHEE. *INTECOMS: Journal of Information Technology and Computer Science*, 7(3), 639–647.
- Dody Firmansyah, M. (2023). Perancangan Web E-Commerce Berbasis Website pada Toko Ida Shoes. In *Journal of Information System and Technology* (Vol. 04, Issue 01).
- Fathonia, G., & Yahfizham, Y. (2024). *Analisis Studi Literatur Penyelesaian Operator Aritmatika Serta Bilangan Bulat Dengan Code Sederhana Pada Bahasa Pemrograman C++*. 2(1), 9–16. <https://doi.org/10.59841/saber.v2i1.604>
- Giawa, D., & Abdy, S. (2024). Sistem Informasi Kepegawaian Berbasis Web Dengan Metode Scrum Pada Kantor Camat Medan Timur. *Jurnal Informatika Press*, 1(2), 15–21.
- Gumilang, A., Aisyiyah, P., & Devi, R. (2023). Pembangunan Aplikasi Pendaftaran Praktikum Berbasis Web Laboratorium Prodi Teknik Informatika Universitas Muhammadiyah Gresik. *Jurnal Nasional Komputasi Dan Teknologi Informasi*, 6(2).

- Hafsari, R., Ryandi, M., Pasadana, R. K., Agustina, F. F., Bagio, Z. J., & Dari, T. W. (2024). RANCANG BANGUN SISTEM INFORMASI PPDB ONLINE PADA MADRASAH ALIYAH SWASTA AL-FATA MENGGUNAKAN BAHASA PEMOGRAMAN LARAVEL. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(2), 2303–2309.
- Mita Qurrota Ayunin. (2023). Analisis Fatwa DSN MUI Nomor 112/DSN-MUI/IX/2017 Tentang Akad Ijarah Terhadap Jasa Layanan Kontraktor (Studi Kasus CV. PH) di Kota Bandung. *Bandung Conference Series: Sharia Economic Law*, 3(2). <https://doi.org/10.29313/bcssel.v3i2.8394>
- Noviantoro, A., Silviana, A. B., Fitriani, R. R., & Permatasari, H. P. (2022). Rancangan Dan Implementasi Aplikasi Sewa Lapangan Badminton Wilayah Depok Berbasis Web. *Jurnal Teknik Dan Science*, 1(2), 88–103.
- Perdama, F., Kartika, D. S. Y., & Permatasari, R. (2024). PENGUJIAN SISTEM INFORMASI MANAJEMEN ASET PADA PERUSAHAAN PERKEBUNAN MENGGUNAKAN METODE BLACK BOX TESTING. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(3), 4011–4016.
- Shahib, M. U., Irawan, B., & Bhakti, R. M. H. (2024). Penentuan Jarak Lokasi UMKM Menggunakan Metode Euclidean Distance Berbasis Sistem Informasi Geografis. *Jurnal Tekno Kompak*, 18(2), 380–391.
- Sri, *, Balqis, P., Balqis, S. P., Ardiansyah, T., Al, A. T., & Siahaan, A. (2024). Perancangan Sistem Penerimaan Siswa Baru SMK Swasta Yapim Biru-Biru Berbasis Web. *Jurnal Riset Sistem Informasi Dan Teknik Informatika*, 2(3). <https://doi.org/10.61132/merkurius.v2i3.103>

Syahputri, K., Irwan, M., & Nasution, P. (2023). Peran Database Dalam Sistem Informasi Manajemen. *Jurnal Akuntansi Keuangan Dan Bisnis*, 1(2), 54–58.

<https://jurnal.ittc.web.id/index.php/jakbs/index>

Wismoaji, J. (2024). I) *Teknik Informatika Politeknik Sawunggalih Aji*. 12(1), 36–49.



BIOGRAFI PENULIS



Wida Aryanti lahir di Tasikmalaya pada 16 Juli 2002, ia merupakan anak kedua dari tiga bersaudara. Wida menyelesaikan Pendidikan mengengahnya di SMA Negeri 6 Tasikmalaya dengan jurusan IPA. Saat ini Wida sedang menempuh Pendidikan di Universitas Bina Sarana Informatika Kampus Kota Tasikmalaya.



Pramudya Saputra lahir di Tasikmalaya pada 03 Oktober 2002, ia merupakan anak Bungsu dari dua bersaudara. Pramudya menyelesaikan Pendidikan mengengahnya di SMK Mitra Batik Tasikmalaya dengan jurusan Teknik Komputer Jaringan. Saat ini Pramudya sedang menempuh Pendidikan di Universitas Bina Sarana Informatika Kampus Kota Tasikmalaya.



Rina Fitri Rusli lahir di Tasikmalaya pada 21 Februari 2002, ia merupakan anak bungsu dari dua bersaudara. Rina menyelesaikan Pendidikan mengengahnya di SMA Negeri 1 Taraju dengan jurusan IPA. Saat ini Rina sedang menempuh Pendidikan di Universitas Bina Sarana Informatika Kampus Kota Tasikmalaya.



Dddy Supriadi, M.Kom. Lahir di Tasikmalaya, 13 Desember 1968. Telah menekuni dan mengajar bidang komputer sejak tahun 1991 Pendidikan terakhir pada Program Studi Magister Ilmu Komputer di STMIK Nusa Mandiri Jakarta. Saat ini sebagai tenaga pendidik pada Universitas Bina Sarana Informatika Program Studi Sistem Informasi Kampus Kota Tasikmalaya dengan jabatan fungsional dosen dan telah mendapatkan setifikasi pendidik sebagai dosen. Beberapa karya yang penulis buat diantaranya pada tahun 2005 membuat sebuah buku yang diterbitkan oleh penerbit Andi Offset Yogyakarta dengan judul “Cara Cepat Membangun Website Dari Nol” dimana Buku ini menjelaskan tentang step by step membangun website dengan mudah dan praktis dimulai dari NOL, Cara Mudah Memahmi Sistem Informasi Manajemen (2019), Penerapan Konsep Dasar Logika Algoritma Dengan Pemrograman PHP (2023). Selain itu penulis juga aktif menulis Karya Ilmiah lainnya.



Iqbal Dzulfiqar Iskandar, lahir di tasikmalaya 13 oktober 1991, S1 universitas siliwangi, dan melanjutkan jenjang pendidikan S2 di STMIK Magister ilmu Komputer Nusa Mandiri jakarta, bidang ke ahlian data mining. Cita cita dulu jadi baja hitam, tetapi sekarang sebagai Dosen pengajar di BSI Tasikmalaya - Bidang keahlian Sistem Teknologi Informasi, Jaringan, Data mining - Jaringan Komputer - Familiar dengan Sistem Operasi

clear OS, Ms. Windows 98, Windows 2000, Ms. Windows XP, Windows7 dan Windows 8. Mampu mengoperasikan komputer, merakit/memperbaiki PC dan menginstall system operasi komputer. Mampu installasi jaringan computer local area. Mampu merancang kebutuhan system dan membuat perangkat lunak. Mampu menggunakan aplikasi Microsoft Office, Software Pembangun Sistem : Turbo Pascal, C++, Visual Basic 6.0, Macromedia Dreamweaver, Adobe Dreamweaver, Java NetBeans, Flash, Packet tracer. Software Desain Grafis : Macromedia Fireworks MX, Adobe Photoshop, Corel Draw. Konfigurasi jaringan computer. Manajemen Database : Microsoft Access dan MySQL. Mampu menggunakan Internet, Browsing, Email dll.



UNIVERSITAS

PROYEK SISTEM INFORMASI

EXPLORA

ESA GALERY, CIAMIS

Sistem informasi explora adalah sistem informasi yang dirancang untuk menyimpan semua dokumen dan media secara terpusat dan terorganisir, serta meningkatkan efisiensi dan keamanan dalam pengelolaan proyek. Pendekatan yang digunakan adalah metode manajemen proyek berbasis Scrum, yang fleksibel dan adaptif terhadap perubahan, serta teknik observasi partisipatif untuk mendapatkan wawasan yang lebih akurat tentang kebutuhan pengguna dan tantangan yang dihadapi.



Wida Aryanti



Rina Fitri Rusli



Pramudya Saputra



Deddy Supriadi, M.Kom



Iqbal Dzulfiqar Iskandar, S.T., M.Kom