

**PENGEMBANGAN WEBSITE *E-SURVEY*  
MENGUNAKAN *LARAVEL* 12 DI FAKULTAS SAINS DAN  
TEKNOLOGI**

**TUGAS AKHIR**

**Diajukan Oleh:**

**DANI RIZQULLAH**

**210705113**

**Mahasiswa Fakultas Sains dan Teknologi  
Program Studi Teknologi Informasi**



**FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI AR-RANIRY  
BANDA ACEH  
TAHUN 2026 M/1447 H**

**LEMBAR PERSETUJUAN**

**PENGEMBANGAN WEBSITE *E-SURVEY*  
MENGUNAKAN *LARAVEL* 12 DI FAKULTAS SAINS DAN  
TEKNOLOGI**

**PROPOSAL TUGAS AKHIR**

Diajukan kepada Fakultas Sains dan Teknologi  
Universitas Islam Negeri AR-Raniry Banda Aceh  
Sebagai Salah Satu Beban Studi Memperoleh Gelar Sarjana (S1)  
Dalam Prodi Teknologi Informasi

Oleh:

**DANI RIZQULLAH**

**210705113**

**Mahasiswa Fakultas Sains dan Teknologi  
Program Studi Teknologi Informasi**

Disetujui Untuk Diseminarkan Oleh:

Pembimbing I



Malahayati, M.T

NIP.198301272015032003

Pembimbing II

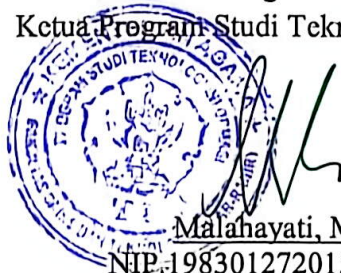


Masrura Mailany, ST., M.T.I

NIP.198903272019031002

Mengetahui

Ketua Program Studi Teknologi Informasi



Malahayati, M.T

NIP.198301272015032003

**LEMBAR PENGESAHAN**

**PENGEMBANGAN WEBSITE *E-SURVEY*  
MENGUNAKAN *LARAVEL 12* DI FAKULTAS SAINS DAN  
TEKNOLOGI**

**TUGAS AKHIR**

Telah Diuji Panitia Ujian Munaqasyah Tugas Akhir  
Fakultas Sains dan Teknologi UIN AR-Raniry Banda Aceh Dinyatakan Lulus  
Serta Diterima Sebagai Salah Satu Beban Studi Program Sarjana (S1)  
Dalam Program Studi Teknologi Informasi

Pada Hari/Tanggal: Rabu 28 Januari 2026  
9 Syaban 1447 H

Di Darussalam, Banda Aceh

Panitia Ujian Munaqasyah Tugas Akhir.

Ketua,

**Malahayati, M.T**  
NIP. 198301272015032003

Sekretaris,

**Masrura Mailany, ST., M.T.I**  
NIP. 198903272019031002

Penguji I,

**Khairan AR, M.Kom**  
NIP. 198607042014031001

Penguji II,

**Dr. Hendri Ahmadian, S.Si., M.I.M**  
NIP. 198301042014031002

Mengetahui,

Dekan Fakultas Sains dan Teknologi  
UIN AR-Raniry Banda Aceh



**Prof. Dr. Jf Muhammad Dirhamsyah, M.T., IPU**  
NIP. 196210021988111001

## LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Dani Rizqullah  
NIM : 210705113  
Program Studi : Teknologi Informasi  
Fakultas : Sains dan Teknologi  
Judul : PENGEMBANGAN WEBSITE *E-SURVEY*  
MENGUNAKAN *LARAVEL 12* FAKULTAS SAINS  
DAN TEKNOLOGI

Dengan ini menyatakan bahwa dalam penulisan tugas akhir ini, saya:

1. Tidak menggunakan ide orang lain tanpa mampu mengembangkan dan mempertanggungjawabkan;
2. Tidak melakukan plagiasi terhadap naskah karya orang lain;
3. Tidak menggunakan karya orang lain tanpa menyebutkan sumber asli atau tanpa izin pemilik karya;
4. Tidak memanipulasi dan memalsukan data;
5. Mengerjakan sendiri karya ini dan mampu bertanggungjawab atas karya ini.

Bila dikemudian hari ada tuntutan dari pihak lain atas karya saya, dan telah melalui pembuktian yang dapat dipertanggungjawabkan dan ternyata memang ditemukan bukti bahwa saya telah melanggar pernyataan ini, maka saya siap dikenai sanksi berdasarkan aturan yang berlaku di Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh. Demikian pernyataan ini saya buat dengan sesungguhnya dan tanpa paksaan dari pihak manapun.

Banda Aceh, 28 Januari 2026

Yang Menyatakan

  
  
Dani Rizqullah

## ABSTRAK

Nama : DANI RIZQULLAH  
NIM : 210705113  
Program Studi : Teknologi Informasi  
Judul : Pengembangan Website *E-survey* Menggunakan *Laravel 12* di Fakultas Sains dan Teknologi.  
Tanggal Sidang : 09 Juli 2026  
Jumlah Halaman : 58  
Pembimbing I : Malahayati, M.T  
Pembimbing II : Masrura Mailany, ST., M.T.I

Pengelolaan data survei di Fakultas Sains dan Teknologi (FST) saat ini masih menghadapi kendala desentralisasi data karena penggunaan *platform* pihak ketiga yang terpisah. Penelitian ini bertujuan membangun aplikasi *e-survey* berbasis web menggunakan framework *Laravel 12* dan *Filament* untuk mengintegrasikan pengelolaan kuesioner serta pelaporan secara terpusat. Sistem ini mengimplementasikan fitur *Role-Based Access Control* (RBAC) untuk keamanan akses dan modul laporan otomatis yang menghitung skor rata-rata (*mean*) secara *real-time*. Metodologi pengembangan menggunakan model *Waterfall* dengan pengujian fungsionalitas melalui metode *Black-box Testing*. Hasil pengujian menunjukkan sistem mampu melakukan validasi data secara akurat, mencegah pengisian ganda (*isFilled check*), serta mempercepat rekapitulasi melalui visualisasi grafik otomatis. Secara keseluruhan, sistem ini efektif mendigitalisasi manajemen survei di lingkungan fakultas, meskipun memerlukan optimasi lanjutan pada ketergantungan koneksi internet untuk rendering grafik.

Kata Kunci: Sistem Informasi, *E-survey*, *Laravel 12*, *Filament*, Otomatisasi Laporan, Statistik.

## KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji syukur penulis panjatkan ke hadirat Allah SWT atas limpahan rahmat, taufik, dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul **“Rancang Bangun Aplikasi *E-survey* Fakultas Sains dan Teknologi Berbasis Web Menggunakan Framework *Laravel 12*”** dengan baik. Shalawat dan salam semoga senantiasa tercurah kepada junjungan Nabi Muhammad SAW, beserta keluarga dan para sahabat beliau yang telah menuntun umat menuju jalan kebenaran.

Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana pada Program Studi Teknologi Informasi, Fakultas Sains dan Teknologi, Universitas Islam Negeri Ar-Raniry Darussalam Banda Aceh. Dalam proses penyusunannya, penulis mendapatkan dukungan, bantuan, serta doa dari berbagai pihak yang telah memberikan semangat, bimbingan, dan motivasi sehingga karya ini dapat terselesaikan.

Dengan penuh ketulusan dan rasa hormat, penulis ingin menyampaikan ucapan terima kasih kepada:

1. **Allah SWT**, atas segala rahmat, hidayah, dan pertolongan-Nya, sehingga penulis diberikan kesehatan, kekuatan, serta kemudahan dalam menyelesaikan setiap tahapan penelitian hingga skripsi ini dapat terselesaikan dengan baik.
2. **Kedua Orang Tua dan Keluarga Tercinta**, Penulis menyampaikan rasa terima kasih yang tulus atas segala doa, dukungan, dan semangat yang tak pernah putus. Kehadiran mereka menjadi sumber kekuatan yang berarti dalam setiap langkah, serta memberikan motivasi yang besar bagi penulis untuk menyelesaikan skripsi ini dengan sebaik-baiknya.
3. **Bapak Dr. Ir. M. Dirhamsyah, M.T., IPU**, selaku Dekan Fakultas Sains dan Teknologi, atas dukungan, fasilitas, dan lingkungan belajar yang mendukung selama penulis menempuh studi.

4. **Ibu Malahayati, M.T.**, selaku Ketua Program Studi Teknologi Informasi sekaligus sebagai **Pembimbing I**, yang telah memberikan bimbingan, arahan, masukan, dan motivasi secara sabar serta penuh dedikasi sejak awal penyusunan hingga tahap akhir skripsi ini. Sikap beliau yang begitu pengertian dan terbuka memberikan kenyamanan tersendiri bagi penulis dalam menjalankan perkuliahan hingga akhir.
5. **Bapak Masrura Mailany, S.T., M.T.I**, selaku **Pembimbing II**, yang telah dengan tulus meluangkan waktu di tengah kesibukan untuk memberikan saran, kritik membangun, dan arahan yang sangat membantu dalam penyempurnaan penelitian ini.
6. **Bapak Khairan AR, M.Kom**, yang telah memberikan bimbingan dan arahan yang sangat berharga dalam masa pengembangan proposal penelitian ini, sehingga peneliti memiliki landasan yang kuat untuk memulai karya ini.
7. **Ibu Cut Ida Rahmadiana, S.Si.**, selaku Admin Program Studi yang telah banyak membantu penulis dalam berbagai keperluan akademik sejak awal perkuliahan hingga saat ini. Beliau adalah sosok yang sangat peduli, ramah, dan selalu berusaha memudahkan urusan mahasiswa dengan penuh kesabaran.
8. Seluruh Bapak dan Ibu Dosen Program Studi Teknologi Informasi, yang telah membekali penulis dengan ilmu pengetahuan, wawasan, dan pengalaman berharga sebagai bekal di masa depan.
9. Seseorang yang selalu hadir memberi dukungan dan bantuan di berbagai kesempatan. Meski namanya tidak tertulis, penulis tetap menyimpan rasa terima kasih atas kontribusi yang telah diberikan.
10. Seluruh teman-teman Prodi Teknologi Informasi angkatan 2021, atas kebersamaan, kerja sama, dan dukungan yang menjadi warna dalam perjalanan studi ini.
11. Diri sendiri, yang telah bertahan, berjuang, dan terus melangkah meskipun rintangan datang silih berganti. Perjalanan menuju tahap ini bukanlah hal yang mudah, penuh dengan lelah, kebingungan, rasa ingin menyerah, dan

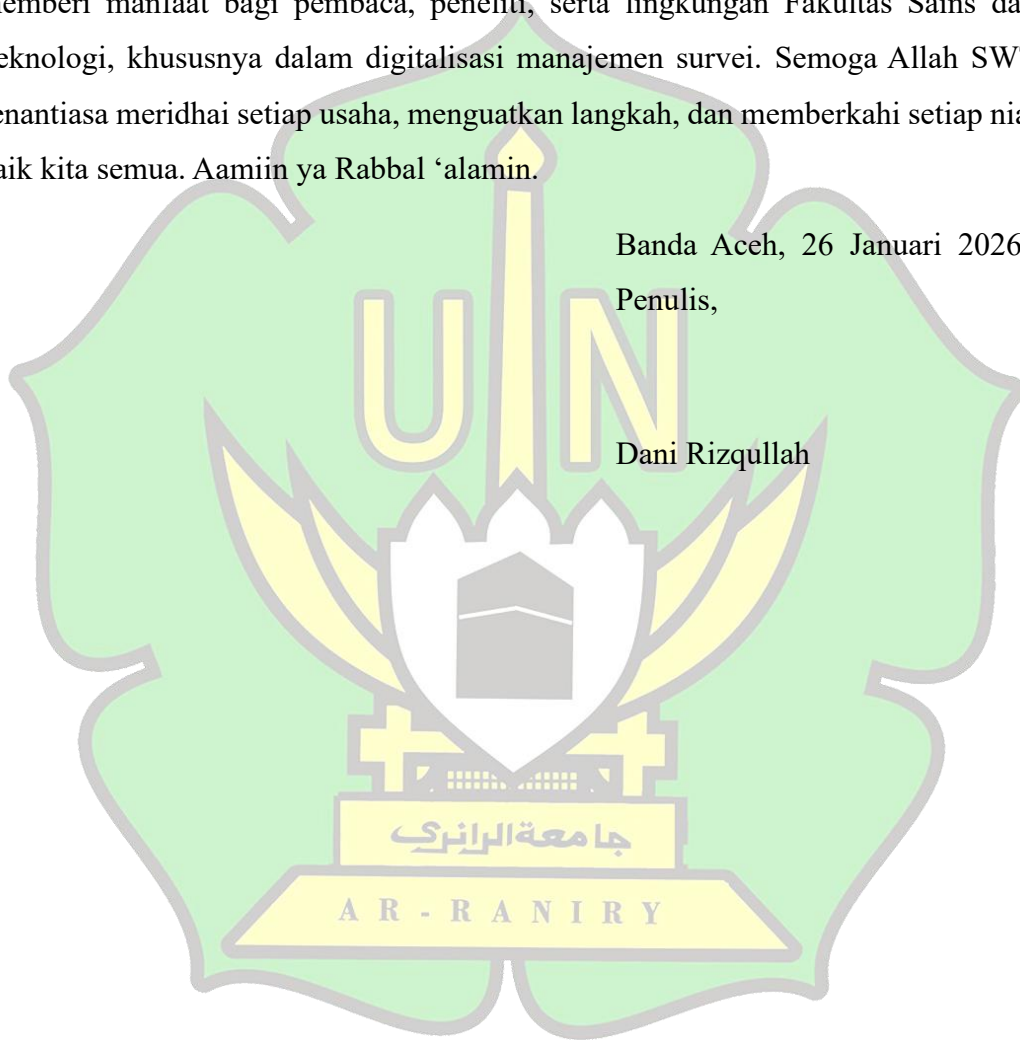
tantangan yang tak terhitung jumlahnya. Namun, setiap langkah kecil yang diambil, setiap usaha yang dikerahkan, dan setiap air mata yang jatuh, semuanya mengantarkan penulis pada titik ini.

Penulis menyadari bahwa skripsi ini masih memiliki kekurangan, baik dari segi isi maupun penyajiannya. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi perbaikan di masa mendatang. Peneliti berharap karya ini dapat memberi manfaat bagi pembaca, peneliti, serta lingkungan Fakultas Sains dan Teknologi, khususnya dalam digitalisasi manajemen survei. Semoga Allah SWT senantiasa meridhai setiap usaha, menguatkan langkah, dan memberkahi setiap niat baik kita semua. Aamiin ya Rabbal 'alamin.

Banda Aceh, 26 Januari 2026

Penulis,

Dani Rizqullah

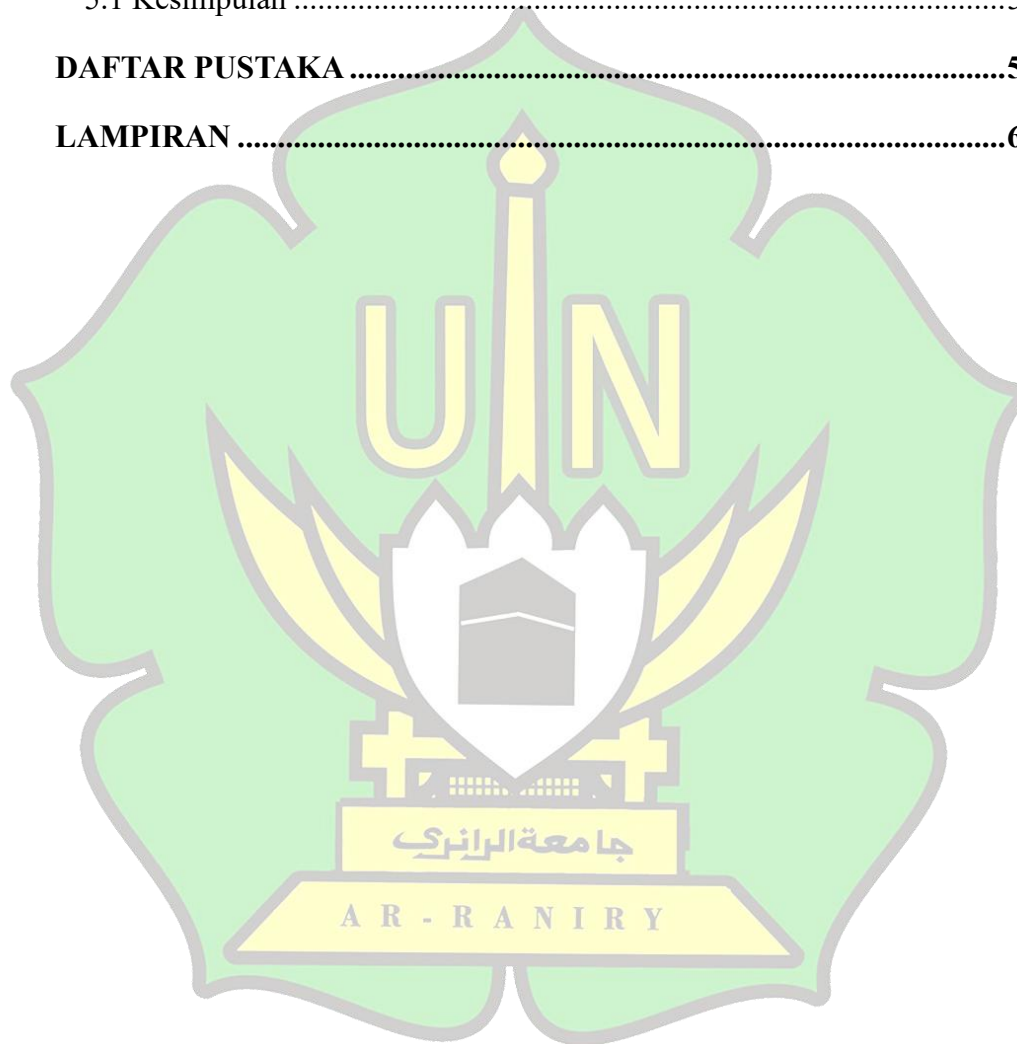


## DAFTAR ISI

<b>KATA PENGANTAR .....</b>	<b>I</b>
<b>DAFTAR GAMBAR.....</b>	<b>VII</b>
<b>DAFTAR TABEL.....</b>	<b>VIII</b>
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1. Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Tujuan Penelitian.....	2
1.4 Batasan Penelitian .....	3
1.5 Manfaat Penelitian .....	3
1.6 Sistematika Penulisan .....	4
<b>BAB II TINJAUAN PUSTAKA.....</b>	<b>5</b>
2.1 Penelitian Terdahulu.....	5
2.2 Teknologi Web .....	8
2.2.1. Definisi dan Konsep Dasar Teknologi Web .....	8
2.3 <i>Hypertext Preprocessor</i> (PHP).....	10
2.3.1. Pengenalan PHP .....	10
2.3.2. Peran PHP dalam Pengembangan Web .....	11
2.4. <i>Laravel</i> .....	12
2.5 <i>Model-View-Controller</i> (MVC) .....	13
1.4 <i>Use case Diagram</i> .....	15
1.5 <i>Activity Diagram</i> .....	15
2.8 <i>Class Diagram</i> .....	16
2.9 Metode <i>Waterfall</i> .....	17
2.9.1. Definisi Metode <i>Waterfall</i> .....	17

2.9.2 Tahapan dalam Metode <i>Waterfall</i> .....	18
2.9.3. Kelebihan dan Kekurangan Metode <i>Waterfall</i> .....	20
2.10. MySQL .....	21
2.10.1. Pengenalan MySQL .....	21
2.10.2. Fitur Unggulan MySQL .....	21
2.10.3. Kelebihan dan Kekurangan .....	21
2.11 Kerangka Berpikir .....	22
<b>BAB III METODE PENELITIAN.....</b>	<b>24</b>
3.1 Tahapan Penelitian .....	24
3.1.1 Identifikasi Masalah .....	24
3.1.2 Pengumpulan Data .....	25
3.1.3 Perancangan sistem.....	27
3.1.4 Implementasi .....	40
3.1.5 Evaluasi .....	40
3.1.6 Pembuatan Laporan.....	41
3.2 Waktu dan Lokasi Penelitian.....	41
3.3 Alat dan Bahan.....	42
3.3.1 Perangkat Keras.....	42
3.3.2 Perangkat Lunak.....	42
<b>BAB IV HASIL DAN PEMBAHASAN .....</b>	<b>43</b>
4.1. Implementasi .....	43
4.1.1. Penerapan Arsitektur Aplikasi .....	43
4.1.2. Implementasi Basis Data .....	43
4.1.3. Implementasi Antarmuka Pengguna (UI).....	44
4.2. Pengujian.....	49
4.2.1. Pengujian Fungsional .....	50

4.2.2. Pengujian Integritas Data dan Keamanan Kritis .....	51
4.3. Pembahasan Hasil .....	53
4.3.1. Analisis Implementasi Fitur Kritis .....	53
4.3.2. Pembahasan Pencapaian Tujuan Penelitian.....	54
<b>BAB V KESIMPULAN .....</b>	<b>55</b>
5.1 Kesimpulan .....	55
<b>DAFTAR PUSTAKA .....</b>	<b>56</b>
<b>LAMPIRAN .....</b>	<b>60</b>



## DAFTAR GAMBAR

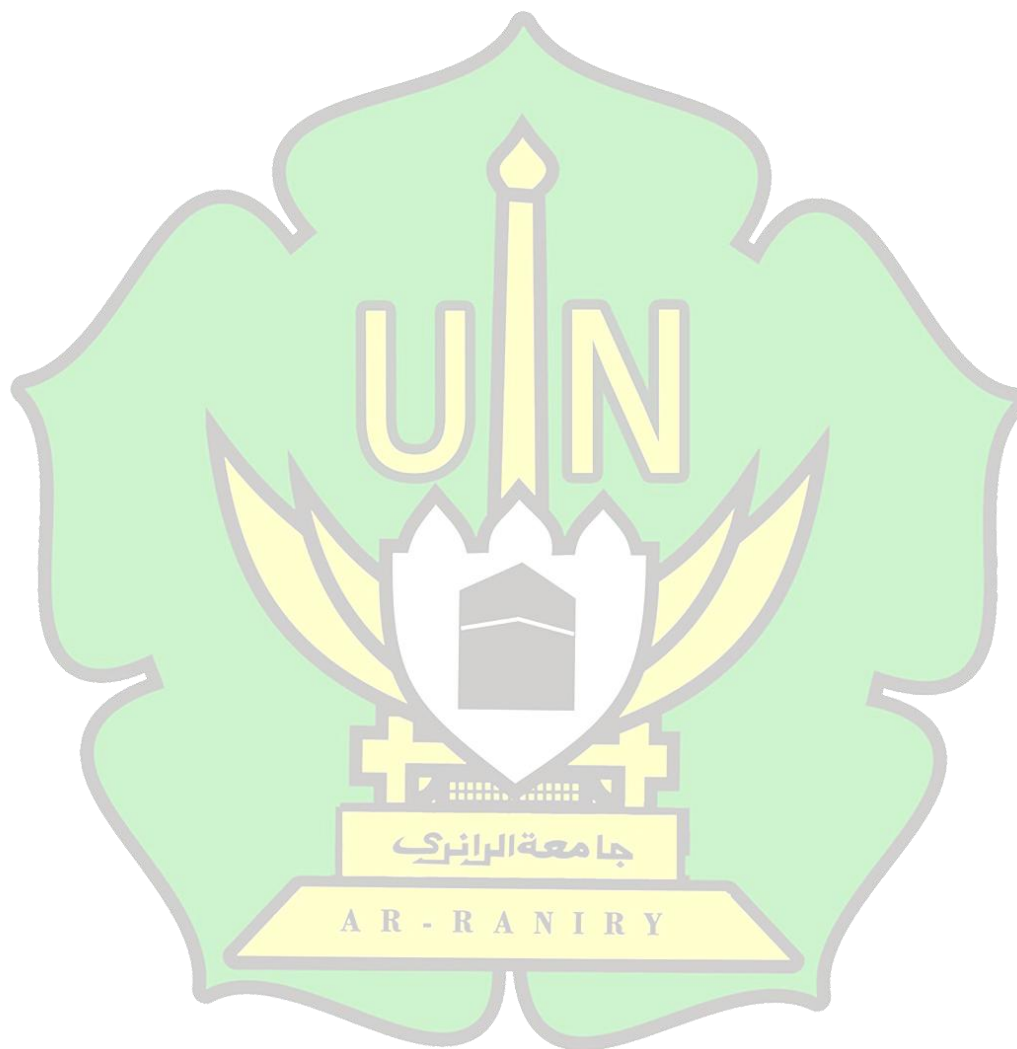
<b>Gambar 2.1</b> Metode <i>Waterfall</i> .....	18
<b>Gambar 2.2</b> kerangka Berpikir .....	23
<b>Gambar 3.1</b> Tahapan Penelitian .....	24
<b>Gambar 3.2</b> Use case Diagram.....	28
<b>Gambar 3.3</b> <i>Activity Diagram</i> Login.....	30
<b>Gambar 3.4</b> <i>Activity Diagram</i> Registrasi .....	31
<b>Gambar 3.5</b> <i>Activity Diagram</i> Isi Survey .....	32
<b>Gambar 3.6</b> <i>Activity Diagram</i> Kelola Konten Survey & Instrument .....	33
<b>Gambar 3.7</b> <i>Activity Diagram</i> Kelola Konten Publikasi Laporan.....	35
<b>Gambar 3.8</b> <i>Activity Diagram</i> Tambah Akun Admin.....	36
<b>Gambar 3.9</b> <i>Class Diagram</i> e-survey .....	39
<b>Gambar 4.1</b> Halaman Login .....	44
<b>Gambar 4.2</b> Halaman Registrasi .....	45
<b>Gambar 4.3</b> Halaman <i>Dashboard</i> Admin.....	46
<b>Gambar 4.4</b> Halaman Form Pembuatan Survei.....	46
<b>Gambar 4.5</b> Halaman tambah pertanyaan .....	47
<b>Gambar 4.6</b> Halaman pengisian survei responden.....	48
<b>Gambar 4.7</b> Halaman pengisian survey mode <i>Read-only</i> .....	48
<b>Gambar 4.8</b> Halaman form publikasi laporan .....	49

جامعة الرانيري

A R - R A N I R Y

## DAFTAR TABEL

<b>Tabel 2.1</b> <i>State of Art</i> Penelitian.....	6
<b>Tabel 3.1</b> Spesifikasi Perangkat Keras .....	42
<b>Tabel 3.2</b> Spesifikasi Perangkat Lunak .....	42
<b>Tabel 4.1</b> Pengujian Fungsionalitas Utama ( <i>Black-box Testing</i> ).....	50
<b>Tabel 4.2</b> Pengujian Integritas Data dan Keamanan Kritis .....	51



# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Pengelolaan data di Fakultas Sains dan Teknologi (FST) memiliki peran penting dalam mendukung administrasi akademik dan proses evaluasi mutu. Data survei yang dikumpulkan menjadi bahan utama untuk melihat kualitas layanan pendidikan secara nyata. Namun, saat ini manajemen survei di lingkungan fakultas masih menghadapi kendala teknis dalam pengumpulan dan pengarsipan datanya.

Saat ini, proses survei di FST masih sering menggunakan platform pihak ketiga seperti *Google Form* atau *Microsoft Forms* secara terpisah-pisah. Hal ini menyebabkan data hasil survei tersebar di banyak akun, tidak terdokumentasi dalam satu pusat data, dan sulit untuk dicari kembali jika sewaktu-waktu dibutuhkan. Selain itu, platform survei gratis lainnya seperti *SurveyMonkey* (versi gratis), *Typeform*, dan *Zoho Survey* juga memiliki batasan pada jumlah responden serta fitur analisis data yang terbatas. Kondisi ini menyulitkan pihak fakultas dalam mengatur hubungan antara data kuesioner, data responden, dan hasil laporan secara terpadu (Saputri et al., 2024).

Oleh karena itu, diperlukan sebuah sistem internal yang dibangun menggunakan *framework Laravel* untuk menyatukan pengelolaan kuesioner dan data responden dalam satu basis data yang terpusat. Pengembangan aplikasi *e-survey* ini dinilai mampu memudahkan admin dalam mengelola instrumen penelitian dan menghasilkan laporan otomatis (Anin et al., 2023). Dengan menggunakan metode *Waterfall*, tahap perancangan sistem dapat dilakukan secara lebih teratur dan jelas. Penggunaan *Laravel* juga dipilih karena kemampuannya dalam menangani banyak pengguna secara stabil (Saputri et al., 2024).

Sistem yang dikembangkan dalam penelitian ini fokus pada penyediaan fitur pengelolaan kuesioner dan visualisasi laporan yang dapat diakses langsung oleh pihak terkait di fakultas. Namun, sebuah aplikasi tentu harus dipastikan dapat berjalan dengan baik sebelum digunakan. Karena itu, diperlukan sebuah tahap pengujian yang sistematis untuk melihat apakah aplikasi sudah sesuai dengan fungsinya.

Dalam penelitian ini, fokus utama diletakkan pada pengujian fungsionalitas menggunakan metode *Black-box Testing*. Pengujian ini dilakukan untuk memastikan bahwa seluruh fitur di dalam sistem, mulai dari pembuatan kuesioner hingga fungsi laporan otomatis, dapat bekerja dengan benar tanpa adanya kesalahan logika. Melalui pendekatan ini, pengembangan *website e-survey* berbasis *Laravel 12* diharapkan dapat menjadi solusi digital dalam manajemen survei di Fakultas Sains dan Teknologi yang teruji secara fungsional.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah diuraikan, khususnya mengenai kendala sentralisasi data survei dan kurangnya standarisasi instrumen di Fakultas Sains dan Teknologi, penelitian ini merumuskan dua permasalahan utama sebagai fokus penelitian:

1. Bagaimana merancang dan membangun aplikasi *e-survey* berbasis *Laravel 12* yang mampu mengintegrasikan pengelolaan instrumen, data responden, dan pelaporan secara terpusat di Fakultas Sains dan Teknologi?
2. Bagaimana hasil pengujian fungsionalitas sistem menggunakan metode *Black-box Testing* dalam memastikan seluruh fitur aplikasi bekerja dengan benar sesuai dengan kebutuhan perancangan yang telah ditetapkan?

## 1.3 Tujuan Penelitian

Tujuan penelitian ini dirumuskan berdasarkan permasalahan yang telah diajukan, yaitu:

1. Merancang dan membangun aplikasi *e-survey* berbasis *Laravel 12* yang mampu menyatukan pengelolaan kuesioner, data responden, dan pelaporan dalam satu sistem terpusat di Fakultas Sains dan Teknologi.
2. Menguji fungsionalitas sistem menggunakan metode *Black-box Testing* untuk memastikan seluruh fitur aplikasi dapat bekerja dengan benar sesuai dengan rencana perancangan yang telah dibuat.

#### 1.4 Batasan Penelitian

Agar penelitian ini memiliki fokus yang jelas, hasil yang terukur, dan dapat diselesaikan sesuai kerangka waktu yang ditetapkan, terdapat beberapa batasan yang ditetapkan dalam ruang lingkup pengembangan sistem, meliputi:

1. Penelitian ini dibatasi pada pengembangan fitur-fitur utama untuk pengelolaan survei, laporan, dan instrumen di lingkungan Fakultas Sains dan Teknologi. Penelitian tidak mencakup integrasi dengan sistem eksternal maupun pengembangan fitur kecerdasan buatan untuk analisis data.
2. Aplikasi yang dikembangkan ditujukan untuk kebutuhan internal di Fakultas Sains dan Teknologi. Kebutuhan spesifik dari institusi luar tidak menjadi fokus dalam penelitian ini.
3. *Framework* yang digunakan dalam pengembangan aplikasi ini adalah *Laravel* versi 12. Penggunaan teknologi lain di luar ekosistem *Laravel* tidak dibahas dalam ruang lingkup penelitian ini.
4. Metode pengembangan sistem menggunakan model *Waterfall* yang dibatasi hingga tahap pengujian fungsionalitas (*Testing*), sehingga tidak mencakup tahap pemeliharaan (*maintenance*) jangka panjang.
5. Evaluasi sistem dibatasi pada pengujian fungsionalitas menggunakan metode *Black-box Testing*. Penelitian ini tidak mencakup pengujian kegunaan (*usability testing*) secara luas kepada responden lapangan atau pengujian performa server di bawah beban ekstrem.

#### 1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan kontribusi dan manfaat bagi berbagai pihak, baik dalam konteks akademik, pengembangan teknologi, maupun lingkungan Fakultas Sains dan Teknologi, antara lain:

1. Bagi Peneliti dan Akademisi: Memberikan wawasan praktis mengenai penerapan arsitektur *Model-View-Controller* (MVC) dalam membangun sistem manajemen survei menggunakan *Laravel* 12. Selain itu, penelitian ini memberikan referensi mengenai prosedur evaluasi sistem menggunakan *Black-box Testing* untuk menjamin kualitas perangkat lunak.
2. Bagi Fakultas Sains dan Teknologi: Menyediakan platform *e-survey* yang telah terverifikasi secara fungsional untuk mendukung pengumpulan data

secara terpusat. Sistem ini membantu meningkatkan akurasi data melalui fitur validasi otomatis dan mempercepat proses pelaporan hasil evaluasi mutu secara *real-time*.

3. Bagi Pengembang Teknologi Informasi: Menjadi acuan dalam mengimplementasikan fitur-fitur kritis seperti *Role-Based Access Control* (RBAC) dan manajemen instrumen (Bank Pertanyaan) menggunakan *Laravel* 12 dan *Filament*.
4. Bagi Institusi: Mendorong digitalisasi proses administrasi akademik yang lebih efisien dan terstandarisasi, sehingga meminimalkan risiko kehilangan data atau kesalahan rekapitulasi manual yang selama ini terjadi.

### 1.6 Sistematika Penulisan

Untuk dapat mengetahui isi penelitian ini, maka secara singkat akan disusun dalam sistematika penulisan sebagai berikut:

1. BAB I PENDAHULUAN  
Bab ini berisi gambaran umum tentang latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan.
2. BAB II TINJAUAN PUSTAKA  
Bab ini memuat materi-materi yang menunjang penelitian ini.
3. BAB III METODE PENELITIAN  
Bab ini menjelaskan berbagai metode yang digunakan untuk penelitian ini dapat selesai.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Penelitian Terdahulu

Beberapa penelitian sebelumnya telah membahas pengembangan sistem informasi berbasis web untuk pengelolaan survei dan data, terutama menggunakan *framework Laravel*. Ardi et al., (2023) dalam penelitiannya merancang aplikasi *SurveyAsia*, yaitu program survei online berbasis web yang digunakan oleh PT. Citiasia Inc. Penelitian ini menekankan kemudahan dalam pengumpulan dan pemantauan data survei secara daring, serta penggunaan *Laravel* dalam struktur pengembangannya memberikan kemudahan pengelolaan data *real-time*.

Selanjutnya, penelitian oleh Nuril Jazilah et al., (2024) merancang sistem survei kepuasan pemangku kepentingan berbasis web di Politeknik Negeri Banyuwangi. Aplikasi ini memungkinkan pengumpulan umpan balik dari berbagai pemangku kepentingan institusi pendidikan secara lebih sistematis dan efisien. Studi ini menunjukkan bahwa sistem yang dibangun mampu mempercepat proses evaluasi dan mendukung penyusunan laporan berbasis data survei.

mengembangkan aplikasi e-Kantin berbasis *Laravel*, yang meskipun fokus utamanya bukan survei, tetapi pendekatan mereka dalam membangun sistem manajemen berbasis *Laravel* memberikan kontribusi terhadap pemahaman penggunaan *Laravel* dalam pengembangan aplikasi web berskala kecil hingga menengah (Raynaldi & Somya, 2023).

Sementara itu, Suarsana et al., (2023) membangun sistem pengukuran Indeks Kepuasan Masyarakat (IKM) berbasis web untuk instansi pemerintahan di Kabupaten Badung. Sistem ini mendukung proses evaluasi layanan publik melalui pengisian survei oleh masyarakat. Pemanfaatan *Laravel* dalam penelitian ini memperlihatkan kemampuannya dalam mengelola alur data survei dan pelaporan dengan struktur yang baik.

Terakhir, penelitian oleh (Sriarthik et al., 2025) berjudul *Survey App: Rating and Feedback System Application*, menunjukkan pengembangan aplikasi yang fokus pada sistem rating dan *feedback* pengguna. Penelitian ini menegaskan

pentingnya sistem umpan balik *real-time* dalam aplikasi berbasis web untuk meningkatkan kualitas layanan dan mendukung pengambilan keputusan yang cepat.

Dari seluruh penelitian tersebut, terlihat bahwa *framework Laravel* banyak dimanfaatkan dalam pengembangan sistem survei karena kemampuannya dalam menangani arsitektur *backend* yang stabil dan terstruktur. Penelitian ini akan melanjutkan arah yang sama dengan fokus pada pengembangan aplikasi *e-survey* di Fakultas Sains dan Teknologi, serta integrasi fitur pengelolaan laporan dan instrumen yang terpusat. Ada pun beberapa penelitian terkait juga dapat di lihat pada table 2.1

**Tabel 2.1** *State of Art* Penelitian

No.	Referensi	Judul Penelitian	Hasil	Perbedaan Kunci
1	(Fadilillah et al., 2025)	Sistem Informasi Survei Kepuasan Masyarakat dengan Framework <i>Laravel</i> pada Pelayanan Publik Pemerintah Daerah	Berhasil menciptakan Sistem Informasi Survei Kepuasan Masyarakat yang berfungsi di lingkungan pelayanan publik menggunakan <i>Laravel</i> .	Penelitian ini tidak berfokus pada Lingkungan Akademik dan tidak mengimplementasikan fitur Manajemen Bank Instrumen untuk standarisasi kuesioner Fakultas.
2	(Fitra & Syukhri, 2023)	Rancang bangun <i>e-survey</i> indeks kepuasan masyarakat terhadap pelayanan pemerintahan desa	Menghasilkan Aplikasi <i>E-survey</i> yang mampu mengukur dan menghitung Indeks Kepuasan Masyarakat (IKM) terhadap pelayanan.	Tidak secara spesifik mengelola Laporan Publikasi Resmi atau mengimplementasikan Skoring Kuantitatif mendalam yang terintegrasi dengan basis data relasional.

3	(Utami et al., 2025)	Perancangan Aplikasi Website Survei Minat Mahasiswa Terhadap Organisasi Kemahasiswaan di Universitas Esa Unggul Kampus Bekasi	Berhasil merancang sistem survei yang memfasilitasi pengumpulan data minat mahasiswa di lingkungan Akademik.	Tidak menggunakan <i>Laravel 12</i> . Tidak fokus pada masalah Integritas Data Transaksional (seperti pencegahan duplicate entry) yang krusial untuk data evaluasi akademik.
4	(Husbi & Ichwani, 2024)	Digitalisasi proses survei badan litbang dan diklat kementerian agama republik indonesia melalui pembuatan aplikasi <i>e-survey</i> berbasis web	Berhasil mendigitalisasi dan memproses data survei secara terpusat di lingkungan institusi besar.	Fokus pada proses, namun kurang menonjolkan fitur <i>Role-Based Access Control</i> (RBAC) per Program Studi dan Filter Data yang dibutuhkan di Fakultas.
5	(Syahira & Muhamad Bahrul Ulum, 2024)	Rancang Bangun Aplikasi Indeks Kepuasan Pasien RS Qadr dengan Integrasi Survei Online Berbasis Website	Berhasil membangun aplikasi Survei Online untuk mengukur Indeks Kepuasan Pasien di lingkungan rumah sakit.	Fokus pada lingkungan non-akademik. Tidak membahas fitur Manajemen Instrumen Survei terpusat dan Manajemen Akun Admin/Super Admin yang kompleks.

## 2.2 Teknologi Web

### 2.2.1. Definisi dan Konsep Dasar Teknologi Web

Teknologi web merupakan sekumpulan alat, bahasa pemrograman, protokol, dan konsep yang digunakan untuk membangun, menyusun, dan menyajikan informasi melalui jaringan internet. Sejak diperkenalkan pertama kali oleh Tim Berners-Lee pada tahun 1991, perkembangan teknologi web telah mengalami transformasi signifikan, dari sekadar dokumen statis menjadi sistem interaktif dan dinamis yang dapat digunakan untuk berbagai kebutuhan, mulai dari komunikasi, edukasi, layanan publik, hingga sistem survei daring (Endra et al., 2021).

Dalam praktik pengembangannya, teknologi web dapat dibagi menjadi dua bagian utama, yaitu *frontend* dan *backend*. *Frontend* mengacu pada bagian dari aplikasi web yang secara langsung berinteraksi dengan pengguna. Pengguna hanya melihat bagian ini, termasuk bagaimana tampilan halaman disusun, bagaimana menu dapat diakses, atau bagaimana formulir dapat diisi dan dikirimkan. Pengembangan *frontend* biasanya menggunakan tiga bahasa utama yaitu *HyperText Markup Language* (HTML), *Cascading Style Sheets* (CSS), dan *JavaScript*. HTML digunakan untuk membentuk struktur dasar halaman web seperti header, paragraf, gambar, dan tautan. CSS digunakan untuk mengatur tata letak dan estetika tampilan seperti warna, ukuran teks, serta posisi elemen-elemen pada halaman. Sementara itu, JavaScript digunakan untuk menambahkan fungsionalitas dinamis yang membuat pengalaman pengguna menjadi lebih interaktif, seperti validasi *input form*, pemanggilan data secara *asynchronous*, dan manipulasi *Document Object Model* (DOM) secara langsung. *Framework* modern seperti *React*, *Vue.js*, atau bahkan *Livewire* dalam ekosistem *Laravel* telah memperkuat kemampuan *frontend* dalam pengembangan aplikasi berbasis web modern (Back4App, 2023; *Laravel*, 2024)

Sementara itu, *backend* merupakan sisi lain dari aplikasi web yang tidak terlihat oleh pengguna, tetapi memainkan peran yang sangat penting dalam pengelolaan data, keamanan, dan logika bisnis. *Backend* bertanggung jawab untuk menangani permintaan dari sisi *frontend*, memproses data, serta menyimpan dan mengambil informasi dari basis data. Teknologi yang umum digunakan dalam pengembangan *backend* antara lain PHP, *Python*, *Node.js*, *Ruby*, dan *Java*. *Laravel*,

sebagai *framework* PHP modern, memberikan kemudahan dalam membangun *backend* yang bersih dan terstruktur melalui pendekatan *Model-View-Controller* (MVC), sistem *routing*, dan *Object Relational Mapping* (ORM) untuk manajemen basis data. Sistem basis data yang sering digunakan adalah *MySQL*, *PostgreSQL*, dan *SQLite*. Semua komponen *backend* ini biasanya berjalan di server, yang menerima permintaan dari browser (*client*) melalui protokol HTTP atau HTTPS dan memberikan respons yang sesuai (Endra et al., 2021)

Arsitektur *client-server* merupakan fondasi dari komunikasi dalam sistem web. Dalam model ini, *client* mengirimkan permintaan ke server, yang kemudian memproses permintaan tersebut dan mengirimkan kembali respons. Protokol komunikasi yang digunakan adalah *Hypertext Transfer Protocol* (HTTP) dan HTTPS (*HTTP Secure*). Perbedaan antara keduanya terletak pada aspek keamanan; HTTPS menggunakan SSL/TLS untuk mengenkripsi data selama transmisi, sehingga informasi pengguna lebih terlindungi dari potensi penyadapan data. Konsep ini menjadi krusial ketika mengembangkan aplikasi yang menangani data sensitif seperti identitas pengguna, informasi survei, atau data keluhan (Patmanthara et al., 2024)

Selain aspek teknis tersebut, pengembangan aplikasi web juga perlu mempertimbangkan prinsip desain responsif. Desain responsif adalah pendekatan desain antarmuka yang memungkinkan tampilan web menyesuaikan diri secara otomatis dengan berbagai ukuran dan jenis perangkat, seperti smartphone, tablet, atau komputer desktop. Dengan desain responsif, pengguna akan tetap mendapatkan pengalaman yang optimal meskipun mengakses aplikasi dari perangkat yang berbeda. Dalam konteks aplikasi survei, hal ini penting agar responden dapat dengan mudah mengisi kuesioner di mana saja tanpa terganggu oleh tampilan yang rusak atau sulit digunakan. Studi yang dilakukan oleh Patmanthara et al., (2024) menunjukkan bahwa penerapan desain web responsif dapat meningkatkan kepuasan dan kenyamanan pengguna secara signifikan, terutama dalam sistem berbasis pembelajaran daring.

Dengan memahami elemen-elemen dasar teknologi web, mulai dari *frontend*, *backend*, protokol komunikasi, hingga prinsip desain responsif, pengembang dapat

merancang sistem berbasis web yang efisien dan efektif. Dalam konteks pengembangan aplikasi *e-survey*, pemahaman ini menjadi penting agar sistem yang dibangun tidak hanya fungsional tetapi juga mudah digunakan, aman, serta dapat diakses dari berbagai perangkat oleh seluruh responden.

## 2.3 Hypertext Preprocessor (PHP)

### 2.3.1. Pengenalan PHP

*Hypertext Preprocessor* (PHP) adalah bahasa *scripting server-side* yang secara khusus dirancang untuk pengembangan aplikasi web. Bahasa ini pertama kali dibuat oleh Rasmus Lerdorf pada awal 1990-an sebagai skrip sederhana untuk personal home page dan pengelolaan formulir. Sejak saat itu, PHP telah berevolusi menjadi bahasa pemrograman yang kuat, fleksibel, dan banyak digunakan di berbagai *platform* web global (Apiag et al., 2023)

Sebagai bahasa *open-source*, PHP memungkinkan pengembang untuk menggunakannya secara bebas tanpa biaya, lisensi, dan melakukan modifikasi sesuai kebutuhan. PHP mendukung berbagai sistem basis data populer seperti *MySQL*, *PostgreSQL*, dan *SQLite*, menjadikannya pilihan tepat untuk membangun aplikasi web dinamis yang interaktif dan efisien dalam interaksi data (Apiag et al., 2023).

Menurut penelitian terbaru, PHP masih mendominasi pasar sebagai bahasa *server-side* utama untuk pengembangan web, dengan lebih dari 75% website dengan bahasa sisi-server menggunakan PHP saat ini (Syarifudin et al., 2025). Popularitas tinggi ini didukung oleh komunitas besar, kompatibilitas dengan berbagai server web seperti *Apache* dan *Nginx*, serta integrasi yang mudah dengan HTML dan *JavaScript*.

PHP juga telah menerapkan paradigma pemrograman modern seperti *Object-Oriented Programming* (OOP) dan struktur *Model-View-Controller* (MVC). *Framework* modern seperti *Laravel* memanfaatkan pendekatan tersebut untuk menghasilkan kode yang lebih modular, terstruktur, dan mudah dipelihara—sehingga cocok digunakan dalam pengembangan sistem *e-survey* dan aplikasi berbasis web lainnya (Perdana et al., 2024)

Dengan basis yang mudah dipelajari, dokumentasi lengkap, dan komunitas yang aktif mengembangkan *library* maupun *tools* PHP, bahasa ini tetap relevan dan banyak digunakan meskipun munculnya bahasa baru dengan fitur yang lebih modern. PHP tetap menjadi fondasi bagi banyak sistem seperti CMS (Wikipedia contributors, 2025), *platform e-commerce*, dan aplikasi administratif perguruan tinggi.

### 2.3.2. Peran PHP dalam Pengembangan Web

PHP memainkan peran penting dalam pengembangan aplikasi web modern, terutama sebagai bahasa pemrograman *server-side* yang mengelola logika bisnis serta interaksi dengan *database*. Ketika pengguna mengakses aplikasi, PHP di server memproses permintaan tersebut, mengambil data dari basis data, dan menghasilkan halaman web dinamis yang kemudian dikirim ke browser pengguna (Wulansari & Kristiana, 2022). Dengan cara ini, aplikasi web menjadi interaktif dan responsif terhadap kebutuhan pengguna.

Keunggulan PHP terletak pada kemampuannya menangani berbagai tipe data dan format, termasuk teks, gambar, dan multimedia. PHP juga mendukung berbagai *framework* populer seperti *Laravel* yang secara signifikan mempermudah pengembangan aplikasi web dengan pola desain yang baik, keamanan lebih tinggi, dan proses pengembangan yang lebih cepat (Ardi et al., 2023)

Selain itu, PHP didukung oleh komunitas pengembang yang besar dan aktif, terutama di Indonesia, yang menyediakan dokumentasi, sumber belajar, dan solusi bagi pengembang ketika menghadapi kendala teknis (Nuril Jazilah et al., 2024). Studi di Politeknik Negeri Banyuwangi menegaskan efektivitas PHP dan *Laravel* dalam membangun sistem survei berbasis web yang sistematis dan mudah diakses (Nuril Jazilah et al., 2024).

Dengan basis *open-source* yang kuat, kemampuan integrasi dengan berbagai sistem *database* seperti *MySQL* dan *PostgreSQL*, serta ekosistem *framework* yang terus berkembang, PHP tetap menjadi pilihan utama untuk membangun aplikasi web, termasuk sistem *e-survey* dan *platform* manajemen data di Indonesia (Raynaldi & Somya, 2023; Wulansari & Kristiana, 2022)

## 2.4. *Laravel*

*Laravel* adalah *framework open-source* berbasis PHP yang dikembangkan oleh Taylor Otwell pada tahun 2011. *Framework* ini dirancang untuk menyederhanakan proses pengembangan aplikasi web dengan menerapkan pola desain MVC, sehingga memisahkan antara logika aplikasi, tampilan, dan data secara terstruktur (Ardi et al., 2023). *Laravel* menawarkan sintaks yang elegan dan fitur-fitur canggih yang memungkinkan pengembang fokus pada pengembangan logika bisnis tanpa terbebani oleh kompleksitas teknis yang sering ditemukan dalam pemrograman web tradisional (Nuril Jazilah et al., 2024a).

Salah satu fitur unggulan *Laravel* adalah *Eloquent ORM* yang memudahkan interaksi dengan *database* tanpa harus menulis *query SQL* secara manual. Dengan *Eloquent*, data dapat diperlakukan sebagai objek PHP, sehingga mempercepat pengembangan dan mengurangi risiko kesalahan dalam pengelolaan *database* (Raynaldi & Somya, 2023). *Laravel* juga menyediakan sistem routing yang fleksibel untuk mengelola URL dan mengontrol alur aplikasi secara efisien.

Untuk pengelolaan tampilan, *Laravel* dilengkapi dengan *Blade*, sebuah *template engine* bawaan yang memungkinkan pembuatan antarmuka pengguna secara dinamis dan mudah dipelajari. *Blade* mendukung sintaks yang sederhana, memungkinkan pengembang membuat halaman web dengan logika kondisional, loop, dan komponen reusable tanpa mengorbankan performa (Wulansari & Kristiana, 2022).

*Laravel* menyediakan *Artisan Console*, sebuah *command-line interface* yang membantu pengembang dalam menjalankan tugas rutin seperti migrasi *database*, pembuatan model, kontrol versi, dan lainnya. Fitur *middleware* memungkinkan kontrol tambahan atas proses permintaan HTTP, sehingga meningkatkan keamanan dan pengelolaan alur data aplikasi (Ardi et al., 2023). Selain itu, *Laravel* mendukung pengujian otomatis, sehingga pengembang dapat memastikan aplikasi berjalan dengan baik dan bebas dari *bug* sebelum dirilis (Nuril Jazilah et al., 2024a).

Keunggulan *Laravel* dalam pengembangan aplikasi web mencakup kemudahan pengkodean, skalabilitas aplikasi, dan keamanan yang kuat. *Framework* ini mempermudah pengembangan aplikasi baik berskala kecil maupun besar dengan

fitur-fitur seperti enkripsi data, perlindungan terhadap serangan *Cross-Site Request Forgery* (CSRF), serta migrasi *database* yang memudahkan perubahan struktur data tanpa kehilangan data (Raynaldi & Somya, 2023). Komunitas *Laravel* yang besar dan aktif di Indonesia turut mendukung pengembang dengan berbagai tutorial, dokumentasi, dan solusi yang mudah diakses (Wulansari & Kristiana, 2022).

Dengan berbagai keunggulan dan dukungan komunitas tersebut, *Laravel* menjadi salah satu *framework* PHP paling populer dan banyak digunakan dalam pengembangan aplikasi web modern, khususnya untuk aplikasi berbasis survei dan sistem informasi.

## 2.5 Model-View-Controller (MVC)

*Model-View-Controller* (MVC) adalah pola arsitektur perangkat lunak yang membagi aplikasi menjadi tiga komponen utama dengan tanggung jawab yang berbeda, yaitu *Model*, *View*, dan *Controller*. Tujuan utama dari penerapan pola ini adalah untuk memisahkan logika aplikasi dari bagian antarmuka pengguna sehingga memudahkan pengembangan, pengelolaan, dan pemeliharaan aplikasi secara lebih efisien dan terstruktur.

Menurut Gibran et al., (2024), "Dengan digunakannya struktur *Model View Controller* (MVC) dapat membuat *Laravel* lebih mudah untuk dipahami dan mempersingkat proses pembuatan web application prototype." Pola ini membagi tugas aplikasi menjadi tiga bagian utama:

*Model* adalah komponen yang bertanggung jawab terhadap pengelolaan data dan logika bisnis aplikasi. *Model* berinteraksi langsung dengan basis data untuk mengambil, menyimpan, memperbarui, dan menghapus data. Dalam *Laravel*, *Model* diwakili oleh *Eloquent ORM*, yang memungkinkan pengembang mengelola data menggunakan objek PHP tanpa perlu menulis query SQL secara manual. Hal ini membuat proses pengembangan lebih produktif sekaligus mengurangi risiko kesalahan dalam pengelolaan data.

*View* merupakan bagian yang menangani tampilan antarmuka pengguna. Tugas *View* adalah menyajikan data dalam format visual yang mudah dipahami oleh pengguna. *Laravel* menggunakan *Blade* sebagai *template engine* bawaannya, yang

menyediakan sintaks sederhana dan efisien untuk membangun antarmuka yang dinamis dan terstruktur. *Blade* juga mendukung fitur seperti template inheritance dan komponen yang mempermudah pengelolaan tampilan yang kompleks.

*Controller* bertindak sebagai penghubung antara *Model* dan *View*. *Controller* menerima *input* atau aksi dari pengguna, memprosesnya sesuai dengan kebutuhan aplikasi, mengelola data melalui *Model*, dan kemudian meneruskan data tersebut ke *View* untuk ditampilkan. *Controller* memastikan bahwa logika aplikasi tetap terpusat dan tidak tercampur dengan bagian tampilan maupun manajemen data.

Penerapan pola MVC dalam *Laravel* sangat membantu pengembang dalam membangun aplikasi yang lebih terorganisir dan modular. Dengan adanya pemisahan tanggung jawab antar komponen, tim pengembang dapat bekerja secara paralel tanpa saling mengganggu. Misalnya, desainer dapat fokus mengembangkan antarmuka pengguna pada *View*, sementara programmer *backend* bekerja pada logika bisnis yang diimplementasikan di *Controller* dan *Model*.

Keunggulan lain dari pola MVC adalah kemudahan dalam proses *debugging* dan *refactoring*. Karena kode dipisah sesuai fungsinya, kesalahan dalam aplikasi dapat dengan mudah ditemukan dan diperbaiki. Selain itu, perubahan pada satu komponen, seperti memperbarui tampilan antarmuka, dapat dilakukan tanpa harus mengubah bagian lain selama struktur data tetap konsisten.

Dalam konteks pengembangan aplikasi berskala besar, seperti sistem informasi akademik, *e-commerce*, atau sistem survei digital, pola MVC menjadi sangat penting untuk menjaga kualitas kode dan mendukung pengembangan berkelanjutan. Struktur ini membantu meningkatkan skalabilitas dan maintainability aplikasi secara keseluruhan.

Dengan demikian, penerapan MVC pada *Laravel* tidak hanya membuat proses pengembangan lebih efisien tetapi juga menghasilkan aplikasi yang lebih mudah untuk dikembangkan, diuji, dan dipelihara sesuai kebutuhan pengguna dan perkembangan teknologi.

#### **1.4 Use case Diagram**

*Use case Diagram* adalah salah satu diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan interaksi antara aktor (pengguna atau sistem eksternal) dengan sistem secara visual. Diagram ini fokus pada kebutuhan fungsional sistem dengan mendokumentasikan fungsi atau layanan (*use case*) yang disediakan oleh sistem bagi aktor yang berinteraksi dengannya. *Use case Diagram* sangat membantu dalam memodelkan persyaratan sistem dari perspektif pengguna, sehingga memudahkan pengembang dan pemangku kepentingan untuk memahami kebutuhan serta batasan sistem (Nuril Jazilah et al., 2024).

Komponen utama dalam *Use case Diagram* meliputi aktor, *use case*, dan hubungan yang menghubungkan keduanya. Aktor adalah entitas eksternal yang berinteraksi dengan sistem, seperti pengguna manusia atau sistem lain. *Use case* merupakan aktivitas atau layanan yang disediakan sistem untuk memenuhi kebutuhan aktor. Hubungan menghubungkan aktor dengan *use case* yang relevan, memperlihatkan interaksi antara keduanya.

Dalam penelitian ini, *Use case Diagram* digunakan untuk memodelkan fungsi utama sistem pengelola survei, laporan, dan instrumen. Diagram ini menggambarkan peran aktor seperti admin dan responden dalam mengakses dan menggunakan sistem, serta berbagai fungsi yang mereka jalankan, seperti pembuatan survei, pengisian survei, dan pengelolaan laporan. Dengan penggunaan *Use case Diagram*, proses pengembangan sistem menjadi lebih terstruktur, dan kebutuhan pengguna dapat diidentifikasi secara jelas sehingga meminimalkan miskomunikasi antara pengembang dan pengguna akhir.

#### **1.5 Activity Diagram**

*Activity Diagram* merupakan jenis diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk memvisualisasikan alur kerja atau aktivitas dalam suatu sistem, termasuk urutan tindakan dan pengambilan keputusan dalam suatu proses bisnis. Diagram ini ideal untuk menggambarkan proses yang kompleks secara grafis, seperti aliran kerja, keputusan bercabang, dan interaksi

antar aktivitas dalam sistem, sehingga mempermudah analisis dan pengembangan sistem (Kurniawan et al., 2023).

Komponen utama dalam *Activity Diagram* meliputi aktivitas (langkah atau tindakan dalam proses), simpul keputusan (*decision point*), alur kontrol (*flow lines*), serta simbol awal dan akhir dari aktivitas.

Dalam konteks penelitian ini, *Activity Diagram* digunakan untuk memodelkan alur kerja sistem pengelola survei, laporan, dan instrumen. Diagram ini menggambarkan langkah-langkah yang dilakukan oleh aktor seperti admin dan responden, mulai dari proses pembuatan survei, pengisian survei, hingga pengelolaan laporan. Dengan *Activity Diagram*, potensi hambatan atau ketidakefisienan dalam proses dapat diidentifikasi sejak tahap desain.

Penggunaan diagram ini juga mendukung kolaborasi tim pengembang dan pemangku kepentingan, karena menyediakan representasi visual yang jelas mengenai cara sistem harus dijalankan dan berfungsi sesuai skenario pengguna.

## **2.8 Class Diagram**

*Class Diagram* adalah salah satu diagram fundamental dalam *Unified Modeling Language* (UML) yang digunakan untuk merepresentasikan struktur statis sebuah sistem perangkat lunak secara menyeluruh. Diagram ini menampilkan kelas-kelas yang ada di dalam sistem, lengkap dengan atribut yang menggambarkan data yang disimpan, serta metode yang menunjukkan perilaku atau fungsi yang dimiliki oleh setiap kelas. Selain itu, *Class Diagram* juga mengilustrasikan hubungan antar kelas seperti asosiasi, agregasi, komposisi, dan pewarisan yang menunjukkan bagaimana objek-objek dalam sistem saling berinteraksi dan berhubungan satu sama lain secara hierarkis maupun fungsional. Menurut (Saputri et al., 2024), penggunaan *Class Diagram* sangat membantu dalam tahap perancangan sistem karena memberikan gambaran yang jelas mengenai struktur dan hubungan komponen dalam perangkat lunak, sehingga memudahkan pengembang dalam proses pemrograman dan perawatan sistem.

Dalam konteks pengembangan sistem pengelolaan survei, laporan, dan instrumen yang dibangun menggunakan *framework Laravel 12*, *Class Diagram*

digunakan untuk menggambarkan kelas-kelas utama yang membentuk sistem tersebut. Beberapa kelas penting yang terdapat dalam diagram meliputi User, yang terdiri dari peran Admin dan Responden, serta kelas-kelas lain seperti *Survey*, *Report*, dan *Instrument*. Masing-masing kelas memiliki atribut yang sesuai dengan data yang harus dikelola, serta metode yang merepresentasikan fungsi yang dapat dijalankan dalam sistem. Dengan adanya hubungan antar kelas yang jelas, seperti asosiasi antara User dengan *Survey* atau *Report*, diagram ini menjelaskan bagaimana komponen sistem saling berinteraksi secara logis dan teknis.

Penggunaan *Class Diagram* tidak hanya membantu tim pengembang untuk memahami bagaimana data dan fungsi terstruktur dalam sistem, tetapi juga memudahkan dalam proses *debugging* dan pengembangan lebih lanjut. Diagram ini menjadi panduan penting dalam implementasi kode program karena menggambarkan *blueprint* yang harus diikuti untuk memastikan bahwa setiap bagian sistem berfungsi dengan benar dan terintegrasi secara baik. Selain itu, *Class Diagram* juga memudahkan pemeliharaan sistem di masa depan karena struktur yang jelas membuat perubahan dan penambahan fitur dapat dilakukan dengan lebih efisien tanpa merusak bagian lain dari sistem.

## **2.9 Metode Waterfall**

### **2.9.1. Definisi Metode Waterfall**

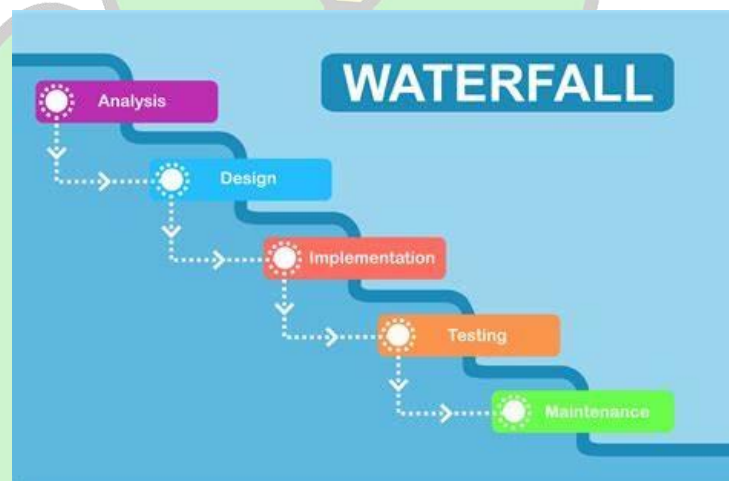
Metode *Waterfall* merupakan salah satu model pengembangan perangkat lunak klasik yang menggunakan pendekatan linear dan berurutan dalam proses pembangunannya. Model ini dinamakan "*Waterfall*" karena menggambarkan aliran proses yang mengalir dari satu tahap ke tahap berikutnya secara sistematis, mirip seperti air terjun. Setiap fase dalam model ini harus diselesaikan secara tuntas sebelum melanjutkan ke fase berikutnya, sehingga tidak memungkinkan untuk kembali ke tahap sebelumnya tanpa melakukan siklus ulang. Model *Waterfall* sangat sesuai digunakan untuk proyek pengembangan yang memiliki kebutuhan yang sudah jelas dan tidak banyak mengalami perubahan selama masa pengembangan.

Menurut Wulansari dan Kristiana (2022), metode *Waterfall* memberikan kerangka kerja yang terstruktur dalam pengembangan perangkat lunak dengan

pembagian tahapan yang jelas. Model ini menekankan pentingnya dokumentasi yang lengkap pada setiap fase sehingga memudahkan pengendalian proyek dan meminimalkan risiko kesalahan. Setiap tahap memiliki keluaran (*output*) yang menjadi dasar dan acuan untuk fase berikutnya, sehingga proses pengembangan menjadi lebih terorganisir dan terukur.

### 2.9.2 Tahapan dalam Metode *Waterfall*

Metode *Waterfall* terdiri dari beberapa tahap yang dilaksanakan secara berurutan, dengan masing-masing tahap memiliki tujuan dan keluaran yang spesifik, dapat di lihat pada gambar 2.1.



**Gambar 2.1** Metode *Waterfall*

a. Analisis Kebutuhan

Pada tahap ini, dilakukan pengumpulan dan analisis secara mendalam terhadap kebutuhan pengguna dan sistem yang akan dikembangkan. Semua kebutuhan harus didokumentasikan secara rinci dan jelas agar dapat menjadi acuan bagi tahap-tahap selanjutnya. Kebutuhan yang tepat dan lengkap sangat krusial agar aplikasi yang dikembangkan sesuai dengan harapan pengguna dan dapat berfungsi dengan baik.

b. Desain Sistem

Setelah kebutuhan dianalisis dan diklarifikasi, tahap berikutnya adalah merancang sistem secara keseluruhan. Desain sistem meliputi perancangan struktur

*database*, antarmuka pengguna (UI), dan logika aplikasi yang akan diimplementasikan. Pada fase ini, rancangan yang matang dan detail akan menjadi panduan utama bagi pengembang dalam membangun aplikasi sehingga dapat menghindari kesalahan dan revisi yang berulang di kemudian hari.

#### c. Implementasi

Tahap ini merupakan proses pembangunan perangkat lunak, di mana para pengembang mulai menulis kode berdasarkan desain yang telah disepakati. Semua komponen dan modul dikembangkan dan diintegrasikan untuk membentuk sebuah sistem yang utuh dan dapat berjalan sesuai spesifikasi.

#### d. Pengujian

Setelah tahap implementasi selesai, sistem yang telah dibangun diuji secara menyeluruh untuk memastikan semua fitur berfungsi dengan baik. Pengujian dilakukan mulai dari pengujian unit, pengujian integrasi antar modul, hingga pengujian sistem secara keseluruhan. Tujuan utama adalah untuk menemukan dan memperbaiki kesalahan atau *bug* agar sistem dapat berjalan stabil dan sesuai dengan kebutuhan.

#### e. Pemeliharaan

Setelah aplikasi resmi diluncurkan, tahap pemeliharaan dimulai dengan melakukan perbaikan atas masalah yang muncul, pembaruan fitur, dan penyesuaian terhadap kebutuhan baru yang mungkin timbul. Pemeliharaan menjadi fase yang sangat penting karena aplikasi harus dapat terus berfungsi dengan baik seiring dengan perubahan lingkungan dan kebutuhan pengguna.

Metode *Waterfall* meskipun sederhana dan terstruktur, memiliki kelemahan yaitu kurangnya fleksibilitas untuk menghadapi perubahan kebutuhan yang signifikan selama proses pengembangan. Oleh karena itu, model ini paling tepat digunakan pada proyek dengan persyaratan yang sudah stabil dan terdefinisi dengan baik sejak awal.

Dengan menerapkan metode *Waterfall*, proses pengembangan menjadi lebih mudah dipantau dan dikendalikan karena setiap tahap memiliki batas waktu dan

keluaran yang jelas. Dokumentasi yang lengkap pada setiap fase juga mendukung komunikasi yang efektif antar tim pengembang dan pemangku kepentingan, sehingga memperkecil risiko miskomunikasi. Adapun dalam penelitian ini, penerapan tahapan *Waterfall* dilaksanakan dengan penyesuaian ruang lingkup yang dibatasi hingga tahap pengujian fungsionalitas sistem, sesuai dengan batasan masalah yang telah ditetapkan pada Bab I.

### **2.9.3. Kelebihan dan Kekurangan Metode *Waterfall***

Metode *Waterfall* menawarkan sejumlah keunggulan yang membuatnya tetap relevan untuk proyek perangkat lunak dengan kebutuhan yang telah jelas dan stabil. Pendekatan ini memiliki struktur yang sistematis, di mana setiap fase mulai dari analisis kebutuhan hingga pemeliharaan dijalankan secara berurutan dan terukur. Hal ini memudahkan pengelolaan proyek karena setiap *deliverable* untuk tiap fase dipersiapkan secara matang, serta dokumentasi yang dihasilkan menjadi sangat komprehensif dan bisa dijadikan acuan untuk pengembangan atau perbaikan di masa depan. Selain itu, karena sifatnya linier, manajer proyek dan tim pengembang bisa fokus menyelesaikan satu fase sebelum beralih ke fase berikutnya tanpa terganggu oleh perubahan mendadak.

Namun demikian, metode *Waterfall* juga memiliki keterbatasan penting. Rendahnya fleksibilitas terhadap perubahan menjadi salah satu kelemahan utama. Jika ada perubahan persyaratan setelah fase analisis selesai, tim harus kembali ke fase sebelumnya yang seringkali menyebabkan penundaan dan kenaikan biaya. Risiko lainnya adalah terkait pengujian yang dilakukan hanya pada tahap akhir; apabila terdapat kekeliruan struktural yang tidak terdeteksi lebih awal, maka perbaikannya bisa sangat rumit dan mahal. Kondisi ini membuat *Waterfall* kurang cocok untuk proyek besar atau yang bersifat dinamis dengan kemungkinan perubahan kebutuhan yang tinggi.

Dalam penelitian yang dilakukan oleh Ananda et al., (2024), metode *Waterfall* diterapkan dalam pengembangan sistem informasi *inventory* berbasis website dan memberikan keuntungan berupa proses pengembangan yang jelas, pengendalian kualitas ketat, serta dokumentasi yang mendukung pemeliharaan.

Namun, mereka juga menyoroti bahwa kurangnya fleksibilitas membuat model ini lebih cocok untuk proyek dengan persyaratan terdefinisi dengan baik sejak awal.

## 2.10. MySQL

### 2.10.1. Pengenalan MySQL

MySQL adalah sistem manajemen basis data relasional (RDBMS) populer berbasis SQL yang bersifat *open-source*. MySQL menjalankan model *client-server* di mana server basis data dapat diakses melalui jaringan oleh berbagai aplikasi klien. *Engine* default InnoDB memberikan dukungan penuh terhadap transaksi ACID (*Atomicity, Consistency, Isolation, Durability*), *foreign key constraint*, *indexing*, serta *stored procedure* dan *View* yang menjaga integritas data dan performa sistem secara keseluruhan. Menurut (Permana et al., 2023), optimalisasi penggunaan indeks sangat memengaruhi kinerja query MySQL, di mana desain *database* yang baik dapat mengurangi waktu eksekusi kueri hingga lebih dari 40 %.

Dalam sistem *e-Survey* berbasis *Laravel 12* ini, MySQL digunakan sebagai basis data utama untuk menyimpan entitas seperti *users* (super admin, admin, responden), *Surveys, Questions, Answers, Reports*, dan *complaints*, dengan relasi antar tabel yang dirancang agar efisien dan konsisten. Struktur ini memungkinkan sistem memproses data survei secara *real-time* dan aman.

### 2.10.2. Fitur Unggulan MySQL

MySQL menawarkan fitur unggulan seperti dukungan *engine* InnoDB yang mendukung transaksi ACID, *foreign key constraints* untuk integritas referensial, serta penggunaan indeks yang efisien guna mempercepat pengambilan data. Fitur *stored procedures* dan *View* mempermudah otomatisasi tugas tertentu dan menyederhanakan akses data kompleks. Kompatibilitasnya dengan PHP dan *Laravel* membuat MySQL menjadi pilihan ideal dalam pengembangan aplikasi web modern.

### 2.10.3. Kelebihan dan Kekurangan

MySQL merupakan sistem manajemen basis data yang banyak digunakan karena memiliki beberapa keunggulan penting. Salah satu kelebihan utama MySQL adalah performanya yang tinggi pada operasi *Create, Read, Update, Delete* (CRUD), terutama jika didukung dengan penggunaan indeks yang tepat. Menurut

(Permana et al., 2023), indeks pada MySQL sangat berpengaruh dalam mempercepat query, khususnya pada operasi join antar tabel yang kompleks dan sering dilakukan dalam aplikasi berbasis web. *Engine InnoDB* pada MySQL juga menjamin keandalan data dengan menerapkan prinsip transaksi *Atomicity, Consistency, Isolation, Durability* (ACID), sehingga data yang disimpan tetap konsisten dan aman meskipun terjadi gangguan. Selain itu, MySQL memiliki dukungan luas terhadap PHP dan *framework Laravel*, serta komunitas yang besar dan dokumentasi lengkap, sehingga memudahkan pengembang dalam integrasi dan pemeliharaan aplikasi.

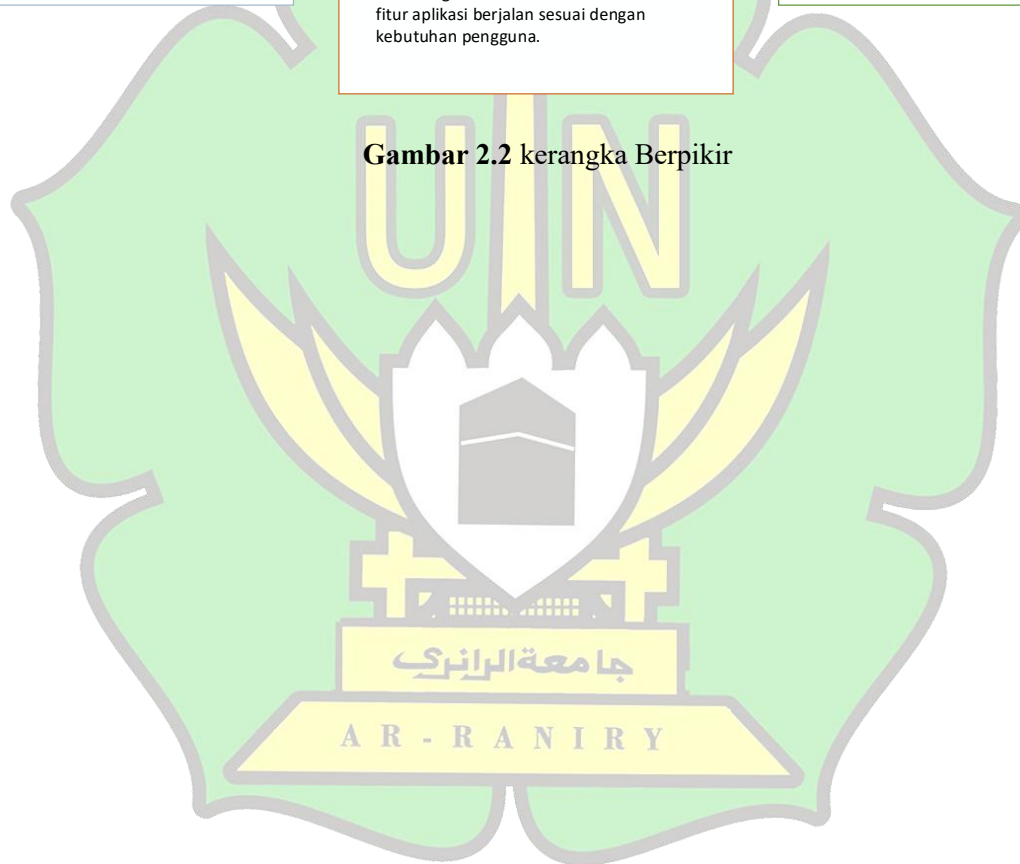
Namun demikian, MySQL juga memiliki beberapa kekurangan yang perlu diperhatikan. Pada sistem dengan trafik tinggi dan akses simultan yang besar, MySQL memerlukan konfigurasi tambahan seperti replikasi dan clustering agar mampu mempertahankan performa yang optimal. Jika tidak dilakukan, maka performa sistem bisa menurun dan menghambat respons aplikasi. Selain itu, query yang kompleks dan melibatkan banyak join akan berjalan lambat apabila desain skema basis data dan indeks tidak dioptimalkan dengan baik. Aspek backup dan recovery juga membutuhkan perhatian khusus agar data tidak kehilangan konsistensi, terutama pada aplikasi yang harus berjalan secara terus-menerus. Dengan demikian, meskipun MySQL memiliki banyak kelebihan, pengelolaan dan konfigurasi yang tepat sangat dibutuhkan untuk menghindari kendala dalam pengoperasian sistem yang berskala besar.

### **2.11 Kerangka Berpikir**

Kerangka Pemikiran adalah struktur paradigma yang digunakan oleh penulis sebagai dasar untuk memahami, menganalisis, dan menyelesaikan masalah. Dalam kerangka ini, berbagai informasi dan konsep dirangkai secara logis menjadi satu kesatuan.



Gambar 2.2 kerangka Berpikir



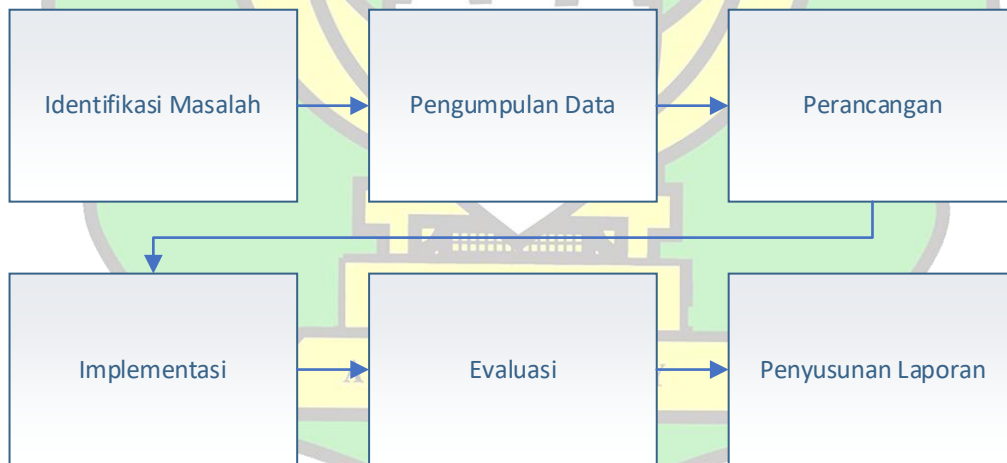
## BAB III

### METODE PENELITIAN

#### 3.1 Tahapan Penelitian

Penelitian ini dilakukan melalui beberapa tahapan utama yang bertujuan untuk menghasilkan sistem pengelolaan survei dan laporan yang efektif dan sesuai kebutuhan pengguna. Tahapan tersebut meliputi identifikasi masalah untuk merumuskan fokus penelitian, pengumpulan data sebagai dasar perancangan sistem, perancangan sistem dengan menggunakan metode *Waterfall*, implementasi pengembangan aplikasi menggunakan *Laravel 12*, evaluasi sistem melalui pengujian dan uji coba pengguna, serta penyusunan laporan penelitian sebagai dokumentasi seluruh proses. Setiap tahapan akan dijelaskan secara lebih rinci pada subbab berikutnya.

Gambar 3.1 memperlihatkan diagram alur tahapan penelitian yang dijalankan dalam penelitian ini.



**Gambar 3.1** Tahapan Penelitian

##### 3.1.1 Identifikasi Masalah

Tahap identifikasi masalah merupakan langkah awal yang dilakukan untuk memperoleh gambaran yang jelas mengenai kondisi dan kebutuhan yang ada di lapangan. Pada penelitian ini, proses identifikasi dilakukan di lingkungan Fakultas

Sains dan Teknologi, dengan fokus pada sistem pengelolaan survei dan pelaporan yang sedang berjalan.

Berdasarkan hasil pengamatan awal, diketahui bahwa pengelolaan survei dan laporan masih sering menggunakan platform pihak ketiga yang terpisah-pisah. Hal ini menyebabkan data hasil survei tersebar di banyak akun, tidak terdokumentasi dalam satu pusat data, dan sulit untuk dicari kembali jika sewaktu-waktu dibutuhkan. Kondisi manajemen data yang belum terpusat ini menyebabkan proses pengolahan data menjadi kurang maksimal dan meningkatkan risiko kehilangan atau ketidaksesuaian data.

Selain itu, mekanisme pelaporan yang belum terpadu menghambat kelancaran distribusi informasi antara pihak pengelola dan pengguna data. Laporan hasil survei seringkali sulit diakses secara cepat oleh pihak yang membutuhkan, sehingga berdampak pada keterlambatan dalam proses evaluasi. Permasalahan utama yang teridentifikasi adalah tidak adanya sistem internal mandiri yang dapat menampung data secara aman serta memudahkan proses pemantauan kinerja.

Berdasarkan hasil analisis tersebut, diperlukan sebuah sistem berbasis web yang terintegrasi untuk mengelola survei, laporan, dan instrumen secara terpusat. Sistem ini diharapkan mampu mempermudah proses pengumpulan, penyimpanan, pengolahan, dan distribusi informasi, sekaligus meminimalkan potensi kesalahan data. Temuan dari tahap ini menjadi landasan utama bagi proses perancangan sistem menggunakan metode *Waterfall* pada tahap berikutnya.

### **3.1.2 Pengumpulan Data**

Pada tahap pengumpulan data, peneliti menggunakan dua metode utama, yaitu observasi dan wawancara, untuk memperoleh gambaran yang jelas mengenai kondisi dan kebutuhan di lapangan terkait sistem pengelolaan survei dan pelaporan di lingkungan Fakultas Sains dan Teknologi.

#### **A. Hasil Observasi (Identifikasi Kendala Proses Lama)**

Observasi dilakukan dengan mengamati secara langsung proses pelaksanaan survei dan pengelolaan laporan yang saat ini berjalan. Hasil observasi menunjukkan adanya beberapa kendala dalam proses manual yang sedang berlangsung:

- a. Penyimpanan Data yang Terpisah: Saat ini, data survei disimpan di berbagai akun platform gratis yang berbeda-beda, sehingga data tidak terkumpul di satu tempat yang resmi milik fakultas.
- b. Risiko Kehilangan Data: Karena file hasil survei biasanya disimpan secara mandiri dalam format Excel di komputer masing-masing unit, terdapat risiko data tersebut sulit ditemukan atau hilang di masa mendatang.
- c. Proses Rekapitulasi Masih Manual: Untuk membuat sebuah laporan hasil survei, pengelola harus mengunduh data terlebih dahulu lalu menghitung skor secara manual di aplikasi pengolah angka, sehingga laporan tidak bisa tersedia secara langsung.

#### **B. Hasil Wawancara (Identifikasi Kebutuhan Sistem Baru)**

Wawancara dilakukan melalui diskusi terarah dengan pihak pengelola data di lingkungan fakultas untuk menggali pengalaman dan kendala dalam pengelolaan survei selama ini. Diskusi ini bertujuan untuk memetakan fungsi-fungsi utama yang harus ada pada aplikasi baru agar sesuai dengan kebutuhan organisasi. Temuan dari hasil diskusi tersebut menghasilkan poin-poin kebutuhan sebagai berikut:

- a. Kebutuhan Aplikasi Mandiri: Diperlukan sebuah sistem informasi internal yang mandiri untuk mengelola seluruh siklus survei secara digital.
- b. Manajemen Instrumen Terpusat: Pengguna mengharapkan adanya fitur Bank Pertanyaan agar instrumen survei dapat disimpan, diarsipkan, dan digunakan kembali secara terstandarisasi.
- c. Pengaturan Otoritas Pengguna: Diperlukan pembatasan hak akses yang jelas antara administrator fakultas dan unit kerja lainnya guna menjamin keamanan dan kerahasiaan data survei.

Berdasarkan temuan dari observasi dan wawancara ini, disimpulkan bahwa data yang diperoleh menjadi landasan utama yang kuat dalam perancangan dan pengembangan sistem pengelolaan survei dan laporan berbasis web pada penelitian ini.

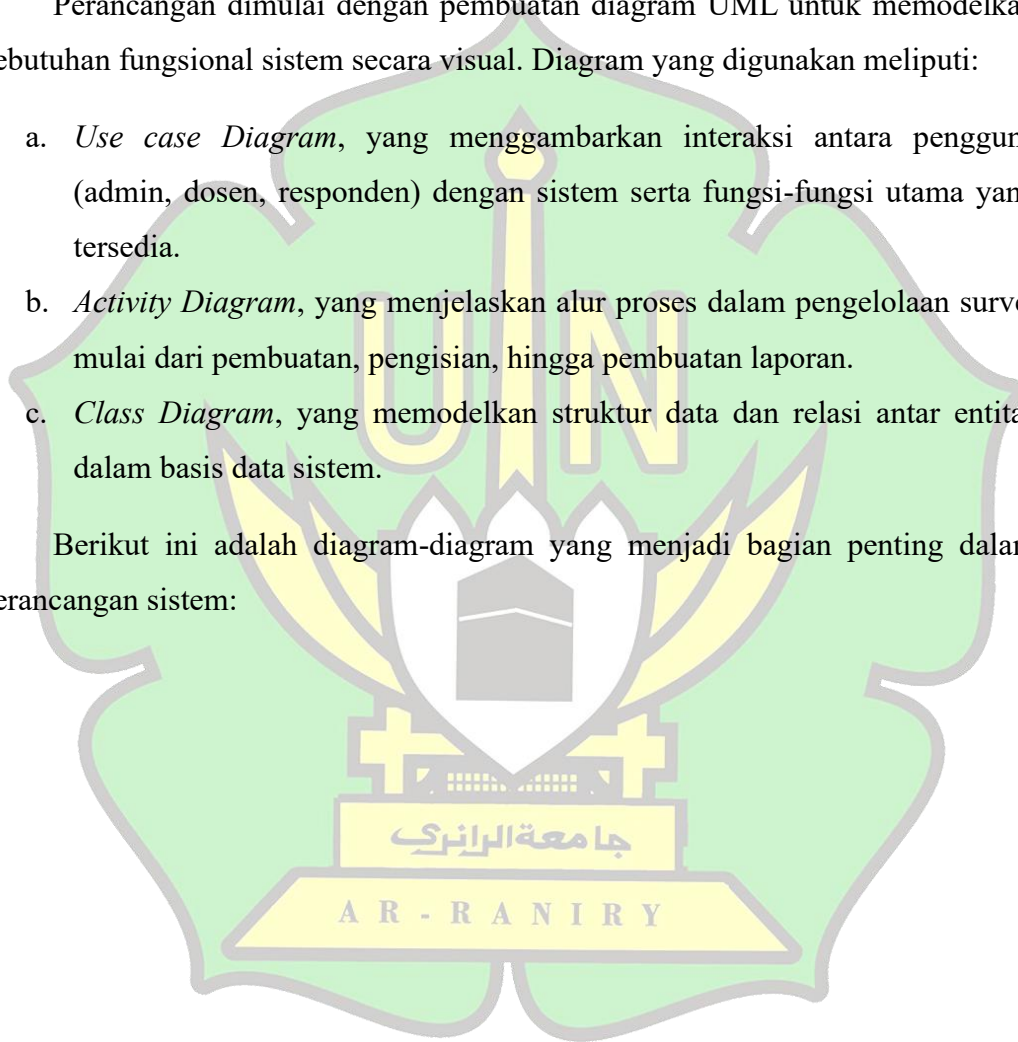
### 3.1.3 Perancangan sistem

Tahap perancangan sistem merupakan proses penting yang bertujuan untuk merancang struktur dan alur kerja aplikasi berdasarkan data dan kebutuhan yang diperoleh pada tahap sebelumnya. Pada penelitian ini, perancangan sistem difokuskan pada pengembangan aplikasi web berbasis *Laravel* 12 untuk mengelola survei, laporan, dan instrumen secara terintegrasi.

Perancangan dimulai dengan pembuatan diagram UML untuk memodelkan kebutuhan fungsional sistem secara visual. Diagram yang digunakan meliputi:

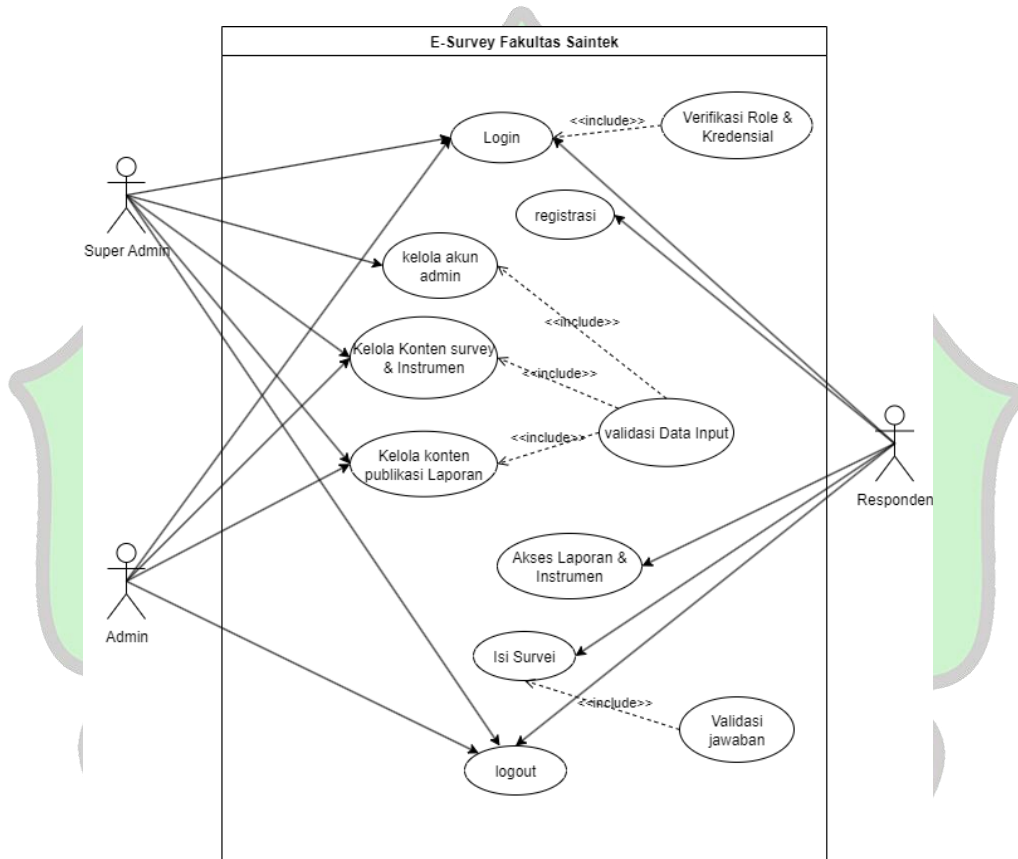
- a. *Use case Diagram*, yang menggambarkan interaksi antara pengguna (admin, dosen, responden) dengan sistem serta fungsi-fungsi utama yang tersedia.
- b. *Activity Diagram*, yang menjelaskan alur proses dalam pengelolaan survei mulai dari pembuatan, pengisian, hingga pembuatan laporan.
- c. *Class Diagram*, yang memodelkan struktur data dan relasi antar entitas dalam basis data sistem.

Berikut ini adalah diagram-diagram yang menjadi bagian penting dalam perancangan sistem:



**a. Use case Diagram**

Diagram *Use case* ini merepresentasikan keseluruhan fungsionalitas Sistem *e-survey* Fakultas Saintek dari sudut pandang tiga aktor utama: Super Admin, Admin, dan Responden. Diagram ini berfungsi sebagai *blueprint* untuk memodelkan kebutuhan fungsional sistem, terutama terkait hak akses dan integritas data. Diagram use case dapat di lihat di Gambar 3.2



**Gambar 3.2 Use case Diagram**

**1. Peran dan Hak Akses Aktor**

Sistem ini membagi pengguna menjadi tiga tingkatan otoritas, yang ditunjukkan melalui garis Asosiasi (garis lurus) ke *Use case* spesifik:

- a. Super Admin: Memiliki hak akses tertinggi. Peran ini unik karena dapat melakukan Kelola Akun Admin, sebuah fungsi yang menjaga keamanan dan privasi data responden dari Admin biasa. Selain itu, Super Admin memiliki semua hak untuk mengelola konten sistem.

- b. Admin: Bertanggung jawab atas pengelolaan konten survei dan laporan. Meskipun memiliki hak akses ke Kelola Konten *Survey* dan Instrumen dan Kelola Konten Publikasi Laporan, sistem akan menerapkan pembatasan internal (berdasarkan Program Studi) agar Admin hanya dapat mengelola data sesuai wewenangnya.
- c. Responden: Merupakan pengguna akhir (mahasiswa/dosen) yang berfokus pada Registrasi Akun, *Login* Sistem, dan fungsi inti aplikasi, yaitu Isi Survei. Responden juga dapat melakukan Akses Laporan dan Instrumen yang telah dipublikasi.

## 2. Fungsionalitas Inti Sistem

Fungsionalitas inti dan persyaratan sistem diwakili oleh *Use case* utama dan didukung oleh *Use case* pendukung (ditandai dengan hubungan <<include>>):

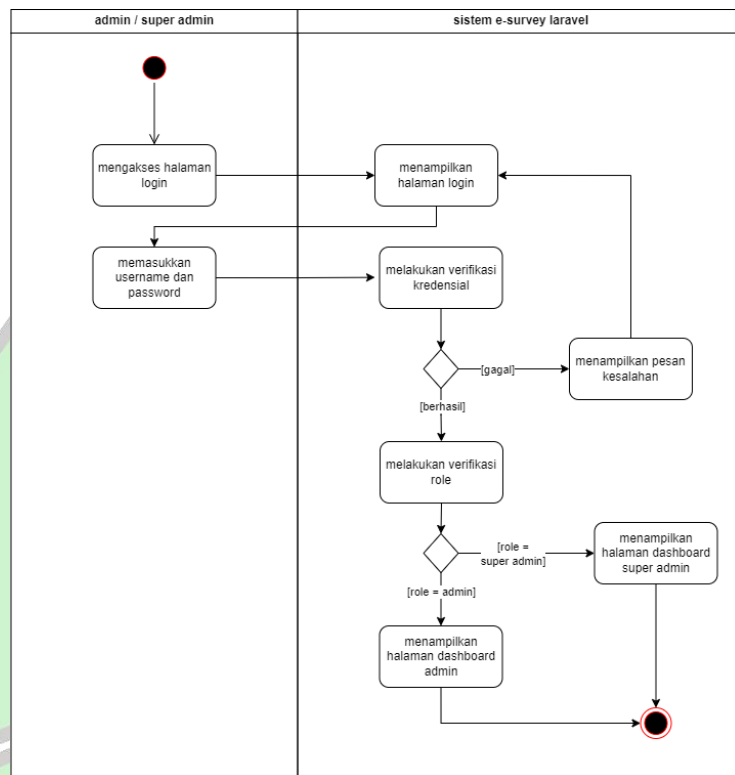
- a. Autentikasi: Semua aktor harus melalui *Login* Sistem, yang secara wajib (<<include>>) menyertakan Verifikasi *Role* & Kredensial. Proses ini menentukan hak akses dan peran pengguna di dalam sistem.
- b. Integritas Data: Semua proses masukan data (registrasi, Kelola Akun Admin, Kelola Konten...) secara wajib (<<include>>) memicu Validasi Data *Input*. Hal ini menjamin bahwa data yang disimpan di *database* selalu lengkap dan sesuai format yang dipersyaratkan oleh sistem (misalnya, memastikan kolom wajib terisi, atau format email benar).
- c. Pengumpulan Data: Fungsi krusial Isi Survei secara wajib (<<include>>) memicu Validasi Jawaban. Validasi ini memastikan bahwa responden telah mengisi semua pertanyaan wajib sebelum data survei dikirim dan disimpan, sehingga meningkatkan kualitas data.

Dengan pemodelan ini, *Use case Diagram* membuktikan bahwa sistem *e-survey* tidak hanya fungsional, tetapi juga memiliki struktur keamanan dan integritas data yang jelas, yang sangat penting untuk aplikasi manajemen informasi akademik.

## b. Activity Diagram

Berikut adalah beberapa *Activity Diagram* yang menggambarkan alur kerja sistem *e-survey*:

### 1. Login

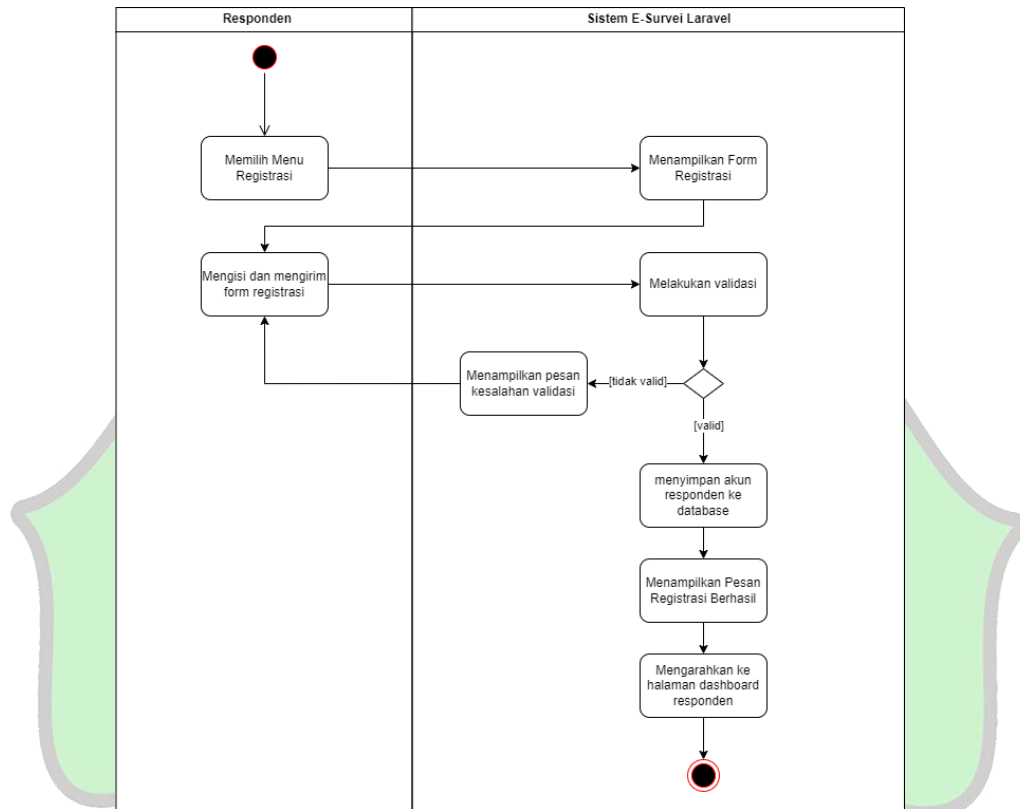


**Gambar 3.3** Activity Diagram Login

Diagram aktivitas *Login* pada Gambar 3.3 memodelkan proses otentikasi yang dilakukan oleh Admin maupun Super Admin untuk mendapatkan akses ke dalam sistem. Alur dimulai ketika pengguna mengakses halaman *Login* dan memasukkan kredensial (nama pengguna dan kata sandi), yang kemudian diverifikasi oleh sistem. Jika proses verifikasi kredensial gagal, sistem akan menampilkan pesan kesalahan kepada pengguna. Namun, jika verifikasi berhasil, sistem akan melanjutkan ke proses otorisasi dengan memeriksa *Role* pengguna yang tersimpan di *database*. Berdasarkan hasil pemeriksaan *Role* ini, sistem mengambil keputusan untuk mengarahkan pengguna ke *Dashboard* yang sesuai: Super Admin akan diarahkan ke Halaman *Dashboard* Super Admin yang memiliki kontrol penuh,

sementara Admin diarahkan ke Halaman *Dashboard* Admin dengan batasan akses yang relevan. Setelah diarahkan, proses *Login* dianggap selesai.

## 2. Registrasi

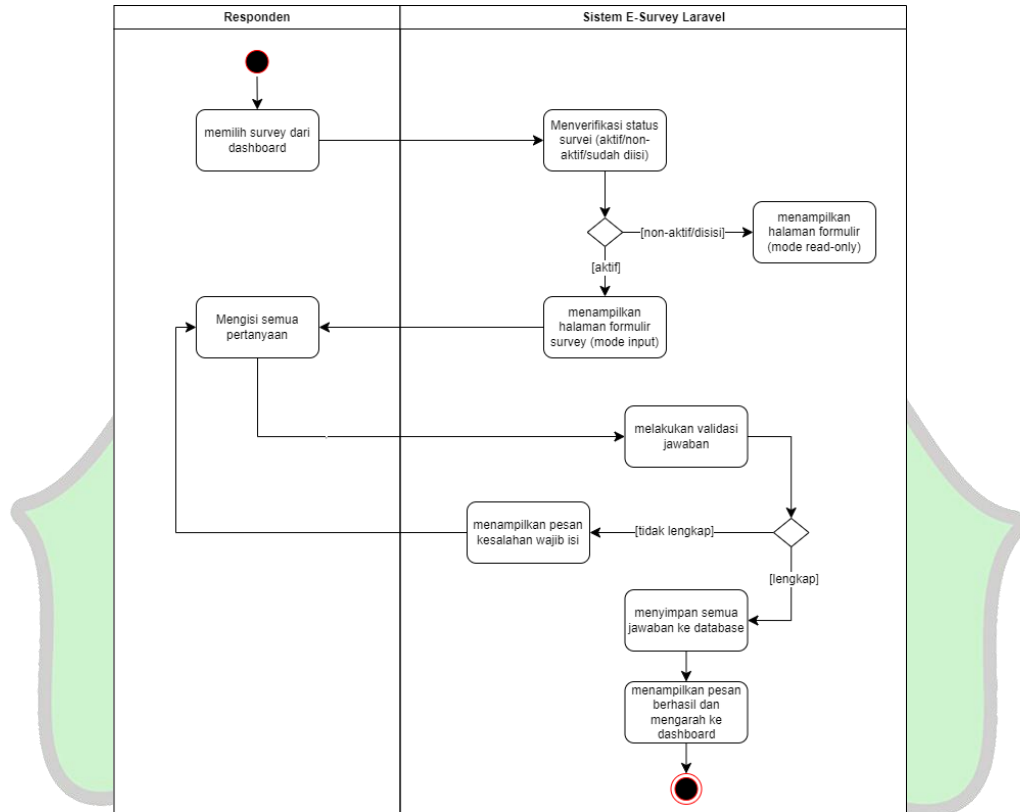


**Gambar 3.4** Activity Diagram Registrasi

Diagram aktivitas Registrasi pada Gambar 3.4 memodelkan alur pembuatan akun baru bagi Responden yang ingin bergabung dengan sistem *e-survey*. Proses dimulai ketika Responden memilih Menu Registrasi, dan sistem merespons dengan menampilkan Form Registrasi. Responden kemudian mengisi dan mengirimkan form registrasi tersebut, yang memicu sistem untuk melakukan validasi data (misalnya, memastikan alamat email unik dan formatnya benar). Jika validasi dinyatakan tidak valid, sistem akan menampilkan pesan kesalahan validasi dan mengembalikan Responden untuk memperbaiki form. Namun, jika validasi dinyatakan valid, sistem akan melanjutkan dengan menyimpan akun responden ke *database*, kemudian menampilkan pesan Registrasi Berhasil. Setelah proses

penyimpanan selesai, sistem secara otomatis akan mengarahkan Responden ke halaman *Dashboard* responden, dan alur dianggap selesai.

### 3. Isi Survei

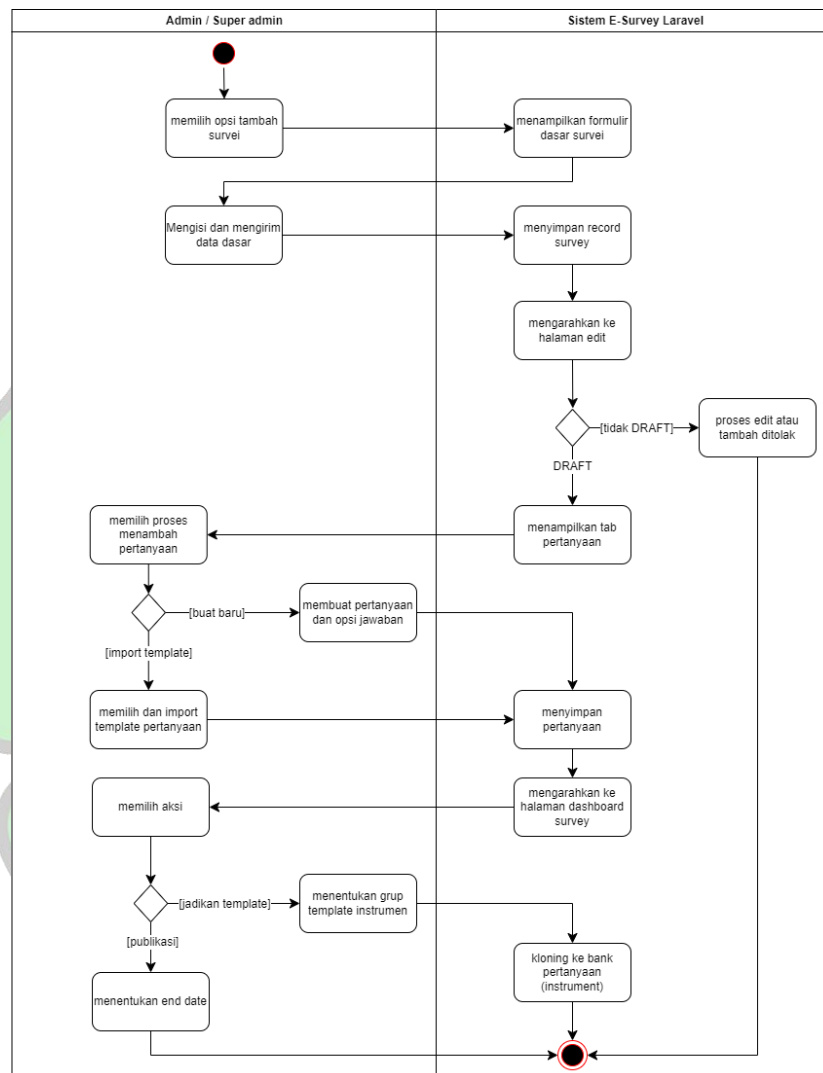


**Gambar 3.5** Activity Diagram Isi Survey

Diagram aktivitas Isi Survei pada Gambar 3.5 memodelkan alur kerja utama Responden saat berinteraksi dengan survei. Alur dimulai ketika Responden memilih survei dari *Dashboard* mereka, yang memicu sistem untuk memverifikasi status survei (apakah survei aktif, non-aktif/tutup, atau sudah pernah diisi). Jika statusnya non-aktif/sudah diisi, sistem akan menampilkan formulir dalam mode *read-only* (hanya-baca), dan alur selesai. Namun, jika survei berstatus aktif, sistem akan menampilkan formulir survei dalam mode *input* penuh. Responden kemudian mengisi semua pertanyaan, dan setelah selesai, sistem akan melakukan validasi jawaban. Jika jawaban dinyatakan tidak lengkap (misalnya, ada pertanyaan wajib yang kosong), sistem akan menampilkan pesan kesalahan wajib isi, dan Responden harus kembali mengisi semua pertanyaan. Jika validasi berhasil (lengkap), sistem

akan menyimpan semua jawaban ke *database*, kemudian menampilkan pesan berhasil dan mengarahkan ke *Dashboard*, menandai selesainya proses pengisian survei.

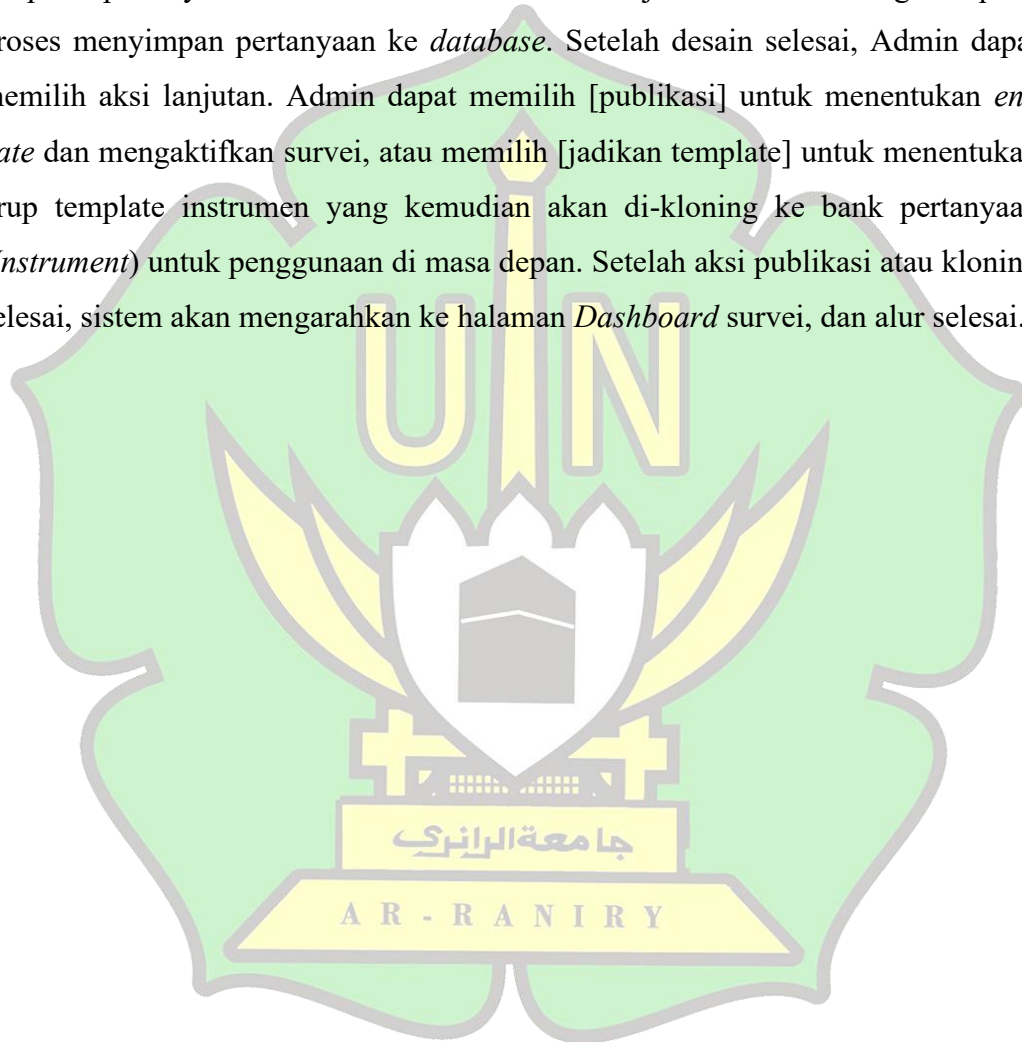
#### 4. Kelola konten *Survey* dan instrumen



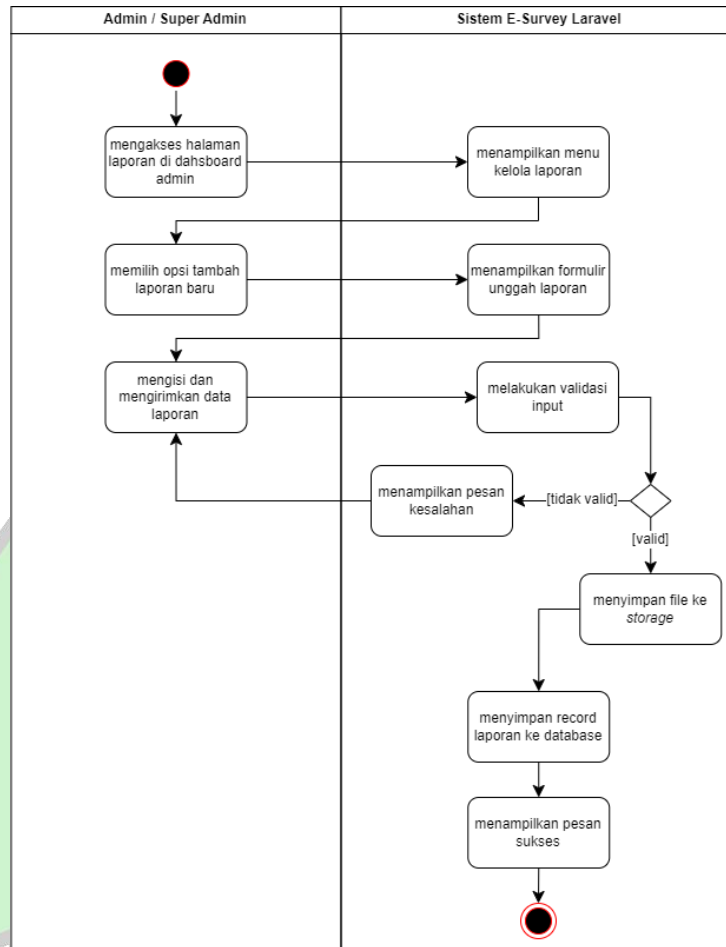
**Gambar 3.6** Activity Diagram Kelola Konten Survey & Instrument

Diagram aktivitas pada Gambar 3.6 memodelkan alur kerja Admin dan Super Admin dalam merancang dan mengelola survei baru, yang juga mencakup fitur bank instrumen. Alur dimulai ketika Admin memilih opsi tambah survei dan mengisi dan mengirim data dasar survei. Sistem akan menyimpan *record* survei tersebut dengan status awal *DRAFT* dan, sebagai optimasi *User Experience*, segera

mengarahkan ke halaman edit. Di halaman ini, sistem akan melalui gerbang keamanan *DRAFT*: jika status survei telah berubah menjadi [TIDAK *DRAFT*] (yaitu, *published* atau *Closed*), proses edit atau tambah pertanyaan akan ditolak, dan alur terputus. Jika statusnya *DRAFT*, Admin dapat memilih proses menambah pertanyaan. Admin dapat memilih jalur [buat baru] untuk membuat pertanyaan dan opsi jawaban secara manual, atau jalur [import template] untuk memilih dan import template pertanyaan dari bank instrumen. Kedua jalur ini akan mengarah pada proses menyimpan pertanyaan ke *database*. Setelah desain selesai, Admin dapat memilih aksi lanjutan. Admin dapat memilih [publikasi] untuk menentukan *end date* dan mengaktifkan survei, atau memilih [jadikan template] untuk menentukan grup template instrumen yang kemudian akan di-kloning ke bank pertanyaan (*Instrument*) untuk penggunaan di masa depan. Setelah aksi publikasi atau kloning selesai, sistem akan mengarahkan ke halaman *Dashboard* survei, dan alur selesai.



## 5. Kelola publikasi laporan



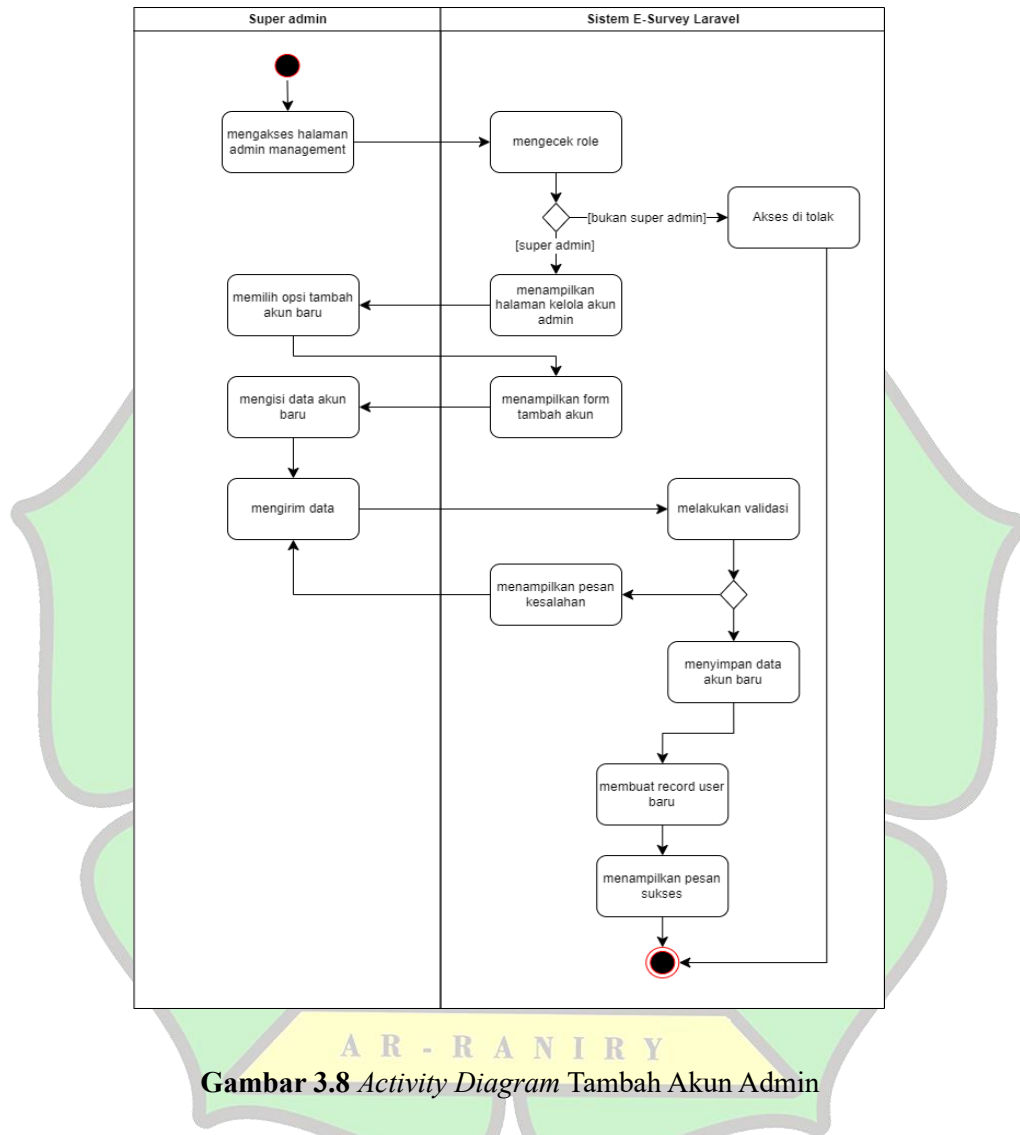
**Gambar 3.7** Activity Diagram Kelola Konten Publikasi Laporan

Diagram aktivitas Kelola Konten Publikasi Laporan pada Gambar 3.7 memodelkan alur kerja Admin dan Super Admin saat mengunggah dokumen laporan hasil survei ke sistem. Alur dimulai ketika Admin atau Super Admin mengakses halaman laporan di *Dashboard*, dan sistem merespons dengan menampilkan menu kelola laporan. Pengguna kemudian memilih opsi tambah laporan baru, yang memicu sistem untuk menampilkan formulir unggah laporan.

Admin mengisi dan mengirimkan data laporan (termasuk *title*, *description*, *file*, dan *tagging target prodi*). Sistem segera melakukan validasi *input* data. Jika hasil validasi [tidak valid], sistem akan menampilkan pesan kesalahan dan Admin harus memperbaiki *input* data. Jika validasi berhasil ([valid]), sistem akan menyimpan *file* ke storage, diikuti dengan menyimpan *record* laporan ke *database*, mencatat

metadata laporan dan path *file*. Setelah penyimpanan selesai, sistem akan menampilkan pesan sukses, dan alur publikasi laporan selesai.

## 6. Tambah akun admin



**Gambar 3.8** Activity Diagram Tambah Akun Admin

Diagram aktivitas Tambah Akun Admin pada Gambar 3.8 memodelkan alur kerja kritis yang hanya dapat diakses oleh Super Admin untuk membuat akun Admin baru, menegaskan kontrol keamanan sistem. Alur dimulai ketika Super Admin mengakses halaman admin management, yang memicu sistem untuk mengecek *Role* pengguna. Jika *Role* adalah [bukan super admin], Akses ditolak, dan alur terputus. Jika *Role* adalah [super admin], sistem akan menampilkan halaman kelola akun admin. Super Admin kemudian memilih opsi tambah akun

baru, dan sistem menampilkan form tambah akun. Setelah mengisi data akun baru (termasuk penentuan *Role* dan detail prodi), Super Admin mengirim data tersebut. Sistem segera melakukan validasi data. Jika data tidak valid, sistem akan menampilkan pesan kesalahan, dan Super Admin harus mengulangi pengisian data. Jika data valid, sistem akan menyimpan data akun baru ke *database* dan membuat *record* user baru. Setelah penyimpanan sukses, sistem menampilkan pesan sukses, dan alur pembuatan akun baru selesai.

### c. *Class Diagram*

*Class Diagram* pada Gambar 3.9 merepresentasikan struktur statis sistem *e-survey*, yang memodelkan delapan (8) entitas utama beserta atribut dan relasi logisnya. Perancangan ini menjamin bahwa sistem memiliki fondasi data yang solid, terstruktur, dan mendukung prinsip akuntabilitas dalam pengelolaan informasi survei.

#### 1. Fondasi Pengguna dan Kontrol Akses

Struktur data dibagi berdasarkan tingkatan fungsionalitas, dimulai dari aktor. Kelas User berfungsi sebagai entitas sentral otentikasi, di mana atribut *Role* (Superadmin, Admin, Responden) dan *program\_studi* (string) menjadi penentu utama hak akses dan penargetan. Relasi Satu-ke-Banyak menjamin bahwa setiap *record* di entitas *Survey*, *Response*, *Laporan*, dan *InstrumentQuestion* dapat ditelusuri kembali ke pencipta (*creator*) atau pengisi (responden).

#### 2. Desain Instrumen dan Siklus Hidup Konten

Kelompok ini mewakili desain survei yang di dalamnya terdapat fitur kontrol kualitas dan daur ulang instrumen.

- a. Kelas *Survey* mengatur siklus hidup konten, yang dikendalikan oleh atribut status (*draft*, *published*, *Closed*). Fitur penargetan audiens dicapai melalui atribut *program\_studi* dengan tipe data JSON.
- b. Kelas *Question* dan *Option* membentuk kerangka instrumen. Setiap *Survey* memiliki relasi wajib (1..\*) terhadap *Question*. Kelas *Option* secara krusial

menyimpan value (*float*), yang merupakan representasi skor numerik dari pilihan jawaban yang berfungsi sebagai kunci untuk analisis kuantitatif.

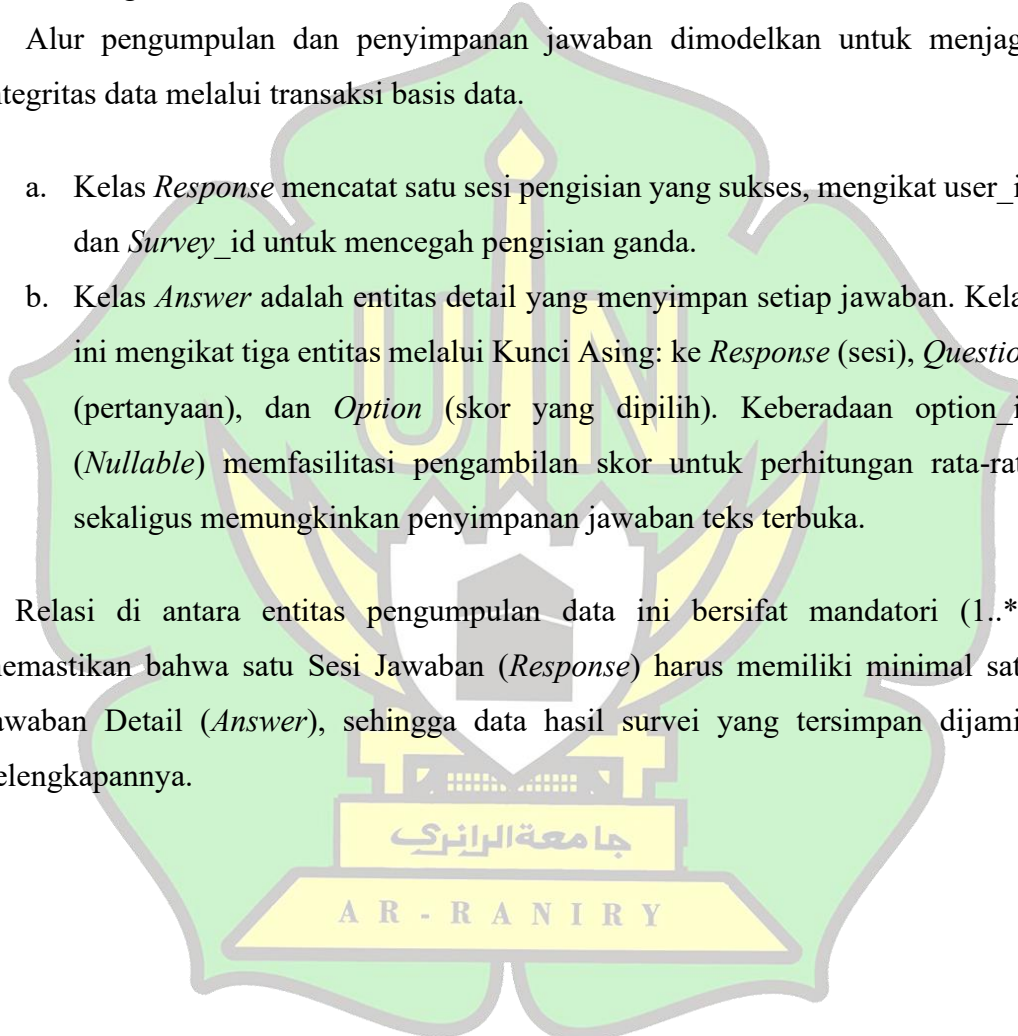
- c. Kelas *InstrumentQuestion* (Bank Template) berfungsi sebagai repositori data daur ulang, menyimpan desain pertanyaan beserta options dalam format JSON untuk diimpor ke survei baru.

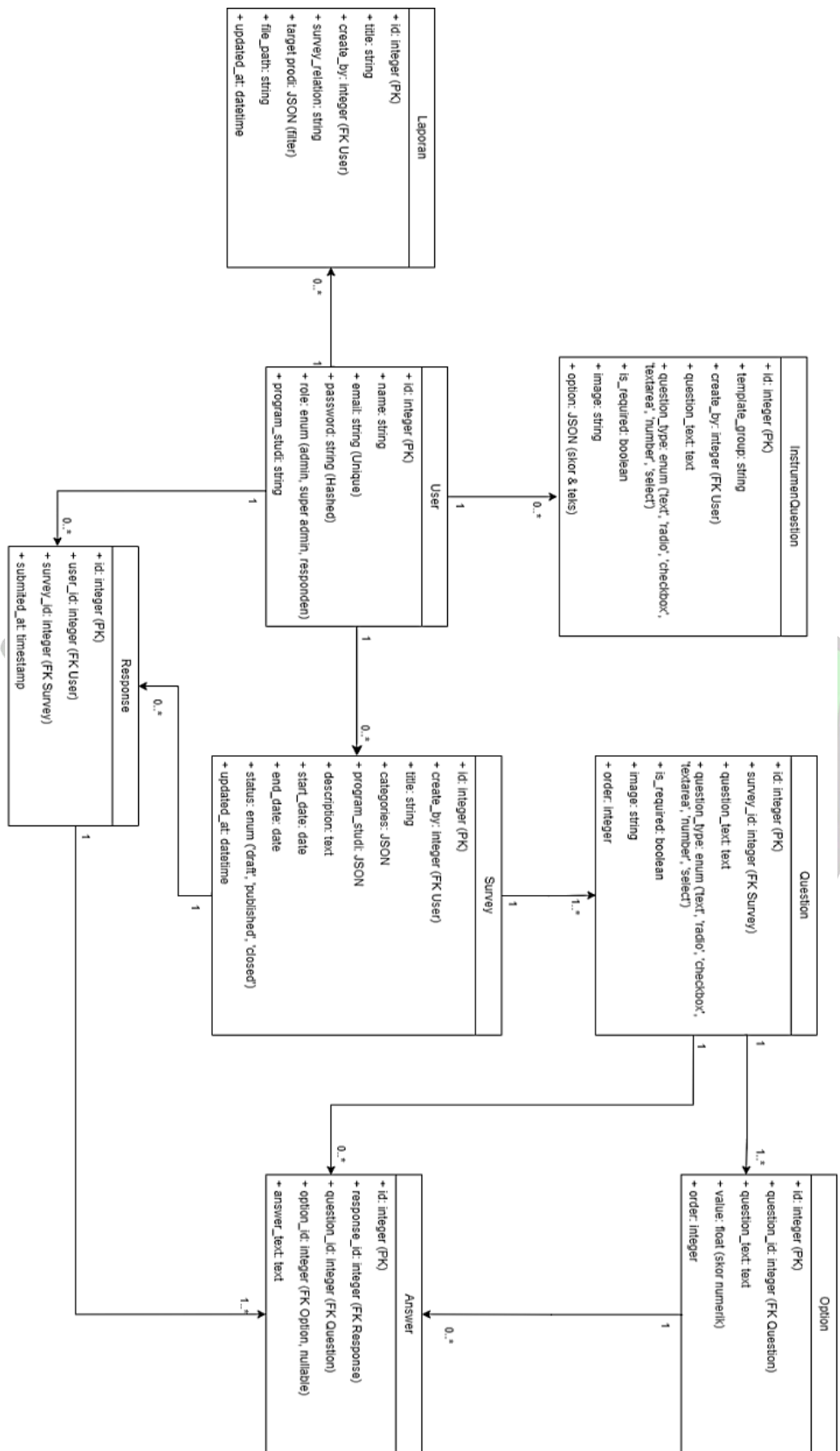
### 3. Integritas Data dan Transaksi Jawaban

Alur pengumpulan dan penyimpanan jawaban dimodelkan untuk menjaga integritas data melalui transaksi basis data.

- a. Kelas *Response* mencatat satu sesi pengisian yang sukses, mengikat *user\_id* dan *Survey\_id* untuk mencegah pengisian ganda.
- b. Kelas *Answer* adalah entitas detail yang menyimpan setiap jawaban. Kelas ini mengikat tiga entitas melalui Kunci Asing: ke *Response* (sesi), *Question* (pertanyaan), dan *Option* (skor yang dipilih). Keberadaan *option\_id* (*Nullable*) memfasilitasi pengambilan skor untuk perhitungan rata-rata sekaligus memungkinkan penyimpanan jawaban teks terbuka.

Relasi di antara entitas pengumpulan data ini bersifat mandatori (1..\*), memastikan bahwa satu Sesi Jawaban (*Response*) harus memiliki minimal satu Jawaban Detail (*Answer*), sehingga data hasil survei yang tersimpan dijamin kelengkapannya.





Gambar 3.9 Class Diagram e-survey

### 3.1.4 Implementasi

Tahap implementasi merupakan tahap di mana rancangan sistem yang telah dibuat pada tahap sebelumnya diwujudkan ke dalam bentuk aplikasi yang berfungsi sesuai dengan kebutuhan yang telah dianalisis. Pada penelitian ini, implementasi dilakukan dengan menggunakan framework *Laravel* versi 12 sebagai kerangka kerja backend. Sedangkan pada sisi *frontend*, halaman pengisian survei untuk responden dikembangkan menggunakan engine *Blade*, sementara dashboard admin dibangun menggunakan *Filament* untuk mempermudah pembuatan antarmuka yang interaktif dan responsif.

Implementasi sistem ini mencakup pengembangan beberapa modul utama, yaitu modul manajemen survei, modul pengisian survei oleh responden, modul pengelolaan pengguna (RBAC), serta modul laporan otomatis dalam bentuk grafik statistik. Selain itu, mekanisme autentikasi diterapkan untuk mengatur hak akses pengguna berdasarkan peran masing-masing, baik sebagai super admin, admin, maupun responden.

Proses implementasi dilakukan secara bertahap dengan mengikuti metodologi pengembangan *Waterfall* yang telah dipilih. Setiap modul yang dibangun diuji secara fungsional untuk memastikan kesesuaian dengan rancangan awal serta untuk menemukan dan memperbaiki kesalahan logika (*bug*) yang muncul selama proses pengembangan. Dengan pelaksanaan tahap implementasi yang terstruktur, dihasilkan sistem pengelolaan survei dan laporan berbasis web yang mampu memenuhi kebutuhan teknis di Fakultas Sains dan Teknologi.

### 3.1.5 Evaluasi

Tahap evaluasi merupakan proses penilaian terhadap sistem yang telah dikembangkan untuk memastikan bahwa aplikasi berfungsi sesuai dengan kebutuhan dan rancangan yang telah ditetapkan. Pada penelitian ini, evaluasi dilakukan dengan pengujian fungsional terhadap seluruh modul sistem, meliputi modul manajemen instrumen, pengisian survei oleh responden, pengelolaan hak akses pengguna (RBAC), serta fitur pelaporan otomatis.

Pengujian dilakukan secara sistematis menggunakan metode *Black-box Testing*, yang berfokus pada hasil masukan (*input*) dan keluaran (*output*) sistem

tanpa memeriksa struktur internal kode. Pengujian ini dirancang untuk menjawab rumusan masalah mengenai ketepatan sistem dalam melakukan validasi data masukan serta kebenaran perhitungan skor otomatis. Selain itu, evaluasi ini bertujuan memastikan bahwa sistem mampu menyajikan rekapitulasi laporan secara otomatis dan terpusat guna mempermudah proses yang sebelumnya dilakukan secara terpisah.

Evaluasi ini juga mencakup simulasi alur kerja aplikasi, mulai dari pembuatan kuesioner hingga publikasi hasil agar sesuai dengan proses administrasi di Fakultas Sains dan Teknologi. Hasil dari tahap ini digunakan untuk memperbaiki kekurangan teknis, sehingga sistem *e-survey* yang dihasilkan terbukti berfungsi dengan baik dan akurat dalam mendukung digitalisasi manajemen survei di lingkungan fakultas.

### **3.1.6 Pembuatan Laporan**

Tahap penyusunan laporan merupakan langkah akhir dalam proses penelitian yang bertujuan untuk mendokumentasikan seluruh kegiatan, hasil, dan analisis yang telah dilakukan. Pada tahap ini, peneliti menyusun laporan skripsi secara sistematis yang mencakup latar belakang, tinjauan pustaka, metodologi penelitian, hasil dan pembahasan, serta kesimpulan dan saran.

Dokumentasi ini sangat penting sebagai bentuk pertanggungjawaban akademik serta sebagai referensi bagi pembimbing dan penguji. Penyusunan laporan juga melibatkan proses revisi berdasarkan masukan pembimbing agar laporan menjadi lebih jelas, lengkap, dan sesuai dengan standar penulisan ilmiah yang berlaku. Dengan tersusunnya laporan penelitian ini, diharapkan seluruh proses dan hasil pengembangan sistem pengelolaan survei dapat dipahami dengan mudah dan memberikan manfaat nyata.

### **3.2 Waktu dan Lokasi Penelitian**

Penelitian ini dilaksanakan dalam jangka waktu dari bulan April 2025 hingga bulan Desember 2025. Proses penelitian dilakukan secara bertahap mengikuti model *Waterfall*, yang meliputi tahap analisis kebutuhan, perancangan, implementasi kode program, hingga tahap pengujian fungsionalitas sistem.

Penelitian ini dilaksanakan pada laboratorium program studi Teknologi Informasi Universitas Islam Negeri Ar-Raniry Banda Aceh.

### 3.3 Alat dan Bahan

Alat yang diperlukan untuk melaksanakan penelitian ini terdiri dari perangkat keras dan perangkat lunak. Analisis mengenai alat dan spesifikasi sistem yang dibutuhkan untuk penelitian ini mencakup:

#### 3.3.1 Perangkat Keras

Perangkat keras yang digunakan berupa satu buah unit Laptop Asus Tuf Gaming A15 dengan spesifikasi yang dipaparkan pada Tabel 3.1

**Tabel 3.1** Spesifikasi Perangkat Keras

Komponen	Spesifikasi
Processor	AMD Ryzen 7 6800H
Ram	16 GB DDR5
Storage	512 GB SSD NVME
Graphics Processing Unit	Nvidia Geforce RTX 3050 Ti 4 GB VRAM

#### 3.3.2 Perangkat Lunak

Perangkat lunak yang dipakai dalam proses implementasi berupa sistem operasi *Microsoft Windows Windows 11 Home Single Language 23H2* juga menggunakan *tools* seperti, *Visual Studio Code*, *Laragon*, dan *Chrome*. *Tools* tersebut akan dipakai dalam penelitian yang akan dilakukan oleh penulis, Dapat dilihat dari Tabel 3.2.

**Tabel 3.2** Spesifikasi Perangkat Lunak

Perangkat Lunak	Version
Windows 11 Home Single Language	23H2
Visual Studio Code	1.101.1
Visual paradigm	17.2
Laragon	6.0
Table plus	6.3.2
Chrome	129.0.6668.72

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1. Implementasi

Tahap implementasi merupakan realisasi dari perancangan logis yang tertuang dalam *Class Diagram* dan *Activity Diagram* pada Bab III. Sistem *e-survey* ini dikembangkan menggunakan framework *Laravel* 12 untuk arsitektur backend dan *Filament/Blade* untuk antarmuka pengguna. Subbab ini menyajikan bukti visual dan fungsional dari penerapan setiap komponen kunci sistem.

##### 4.1.1. Penerapan Arsitektur Aplikasi

Sistem ini dikembangkan dengan mengadopsi pola arsitektur *Model-View-Controller* (MVC). Controller berfungsi sebagai jembatan yang memproses permintaan (*request*), menjalankan logika bisnis, serta melakukan otorisasi peran (*role check*) sesuai *Activity Diagram*. Model diimplementasikan menggunakan Eloquent ORM untuk mengelola data, sementara View (*Filament/Blade*) bertugas menampilkan antarmuka dinamis kepada pengguna.

##### 4.1.2. Implementasi Basis Data

Struktur basis data diimplementasikan berdasarkan *Class Diagram* (Gambar 3.9) melalui Migration *Laravel*, menggunakan MySQL sebagai RDBMS. Penerapan ini menitikberatkan pada integritas data dan logika bisnis kompleks:

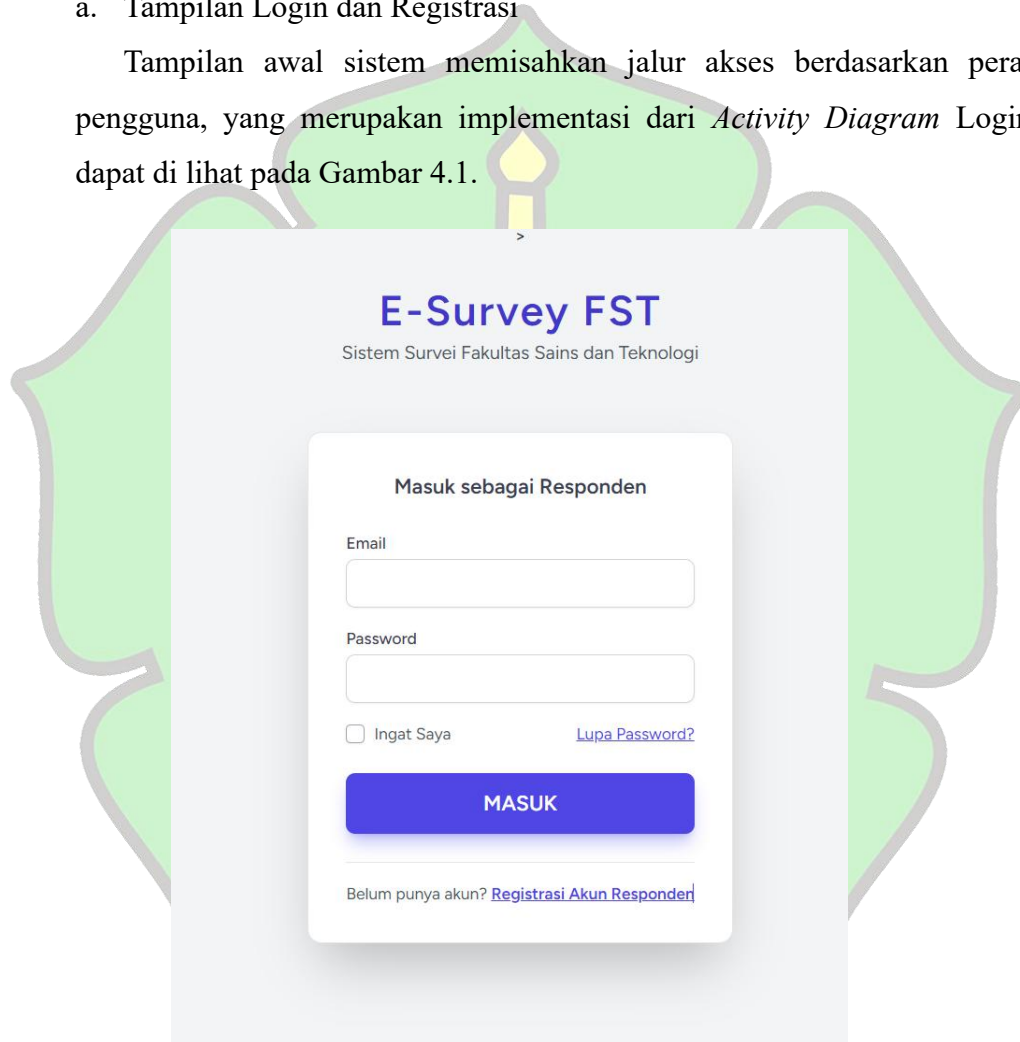
1. Penerapan Relasi Kritis: Semua relasi Satu-ke-Banyak diimplementasikan dengan Foreign Key untuk menjamin integritas referensial.
2. Implementasi Field JSON Kunci: Atribut yang mendukung tagging dan targeting diimplementasikan menggunakan tipe data JSON di MySQL. Ini mencakup *program\_studi* (di tabel *surveys* dan *laporans*) dan *options* (di tabel *instrument\_questions*).
3. Implementasi Skoring: Kunci skoring terletak pada kolom *option\_id* di tabel *Answers*, yang merupakan *Foreign Key* ke tabel *options*. Hal ini memungkinkan sistem untuk mengambil nilai skor numerik (*value float*) dari *options* secara langsung saat perhitungan rata-rata.

### 4.1.3. Implementasi Antarmuka Pengguna (UI)

Subbab ini menyajikan visualisasi sistem yang merupakan wujud nyata dari perancangan *Activity Diagram* pada Bab III. Antarmuka pengguna dikembangkan untuk memfasilitasi interaksi yang mudah bagi Admin dan Responden, sekaligus menerapkan kontrol akses dan integritas data pada sisi *frontend*.

a. Tampilan Login dan Registrasi

Tampilan awal sistem memisahkan jalur akses berdasarkan peran pengguna, yang merupakan implementasi dari *Activity Diagram* Login, dapat di lihat pada Gambar 4.1.



**Gambar 4.1** Halaman Login

Gambar 4.1 memperlihatkan antarmuka Login yang berfungsi sebagai gerbang utama sistem. Tampilan ini mengakomodasi kredensial dari semua peran (admin, superadmin, responden) dan memicu proses Verifikasi Role & Kredensial sebelum pengguna diarahkan ke dashboard yang sesuai.

**E-Survey FST**  
Pendaftaran Akun Responden

Nama Lengkap  Email

Program Studi/Unit Kerja  
Pilih Prodi Anda

Password  Konfirmasi Password

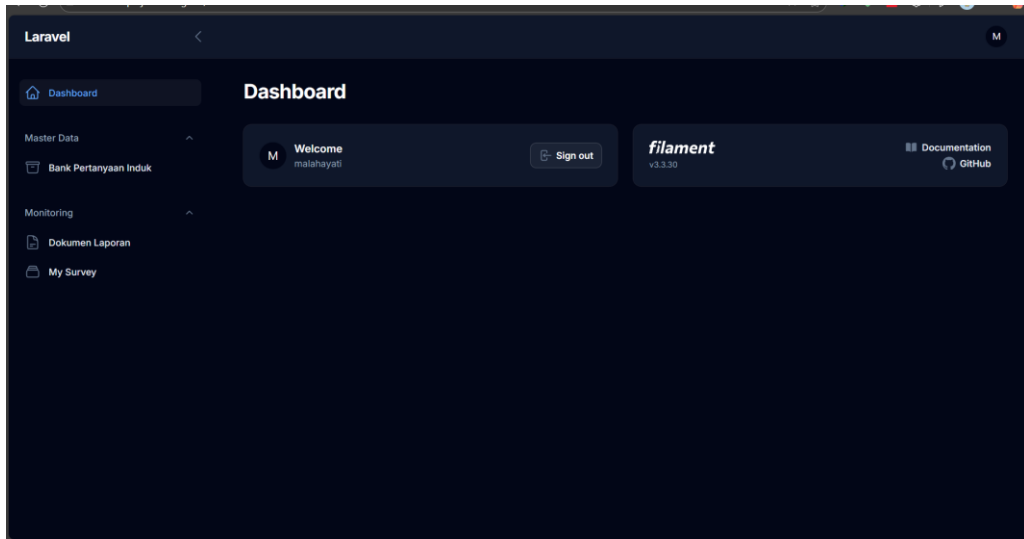
[Sudah punya akun? Masuk](#) **REGISTRASI**

**Gambar 4.2** Halaman Registrasi

Gambar 4.2 menunjukkan antarmuka pendaftaran yang khusus diperuntukkan bagi Responden. Implementasi ini mengharuskan Responden untuk mengisi program\_studi (di mana data ini menjadi penentu konten survei dan laporan yang relevan), sesuai dengan logika awal yang diatur dalam *Activity Diagram* Registrasi.

b. Tampilan Dashboard Admin (*Filament*)

Dashboard Admin diimplementasikan menggunakan *Filament* PHP, yang secara otomatis menerapkan *Role-Based Access Control* (RBAC) untuk membatasi fungsionalitas berdasarkan peran.

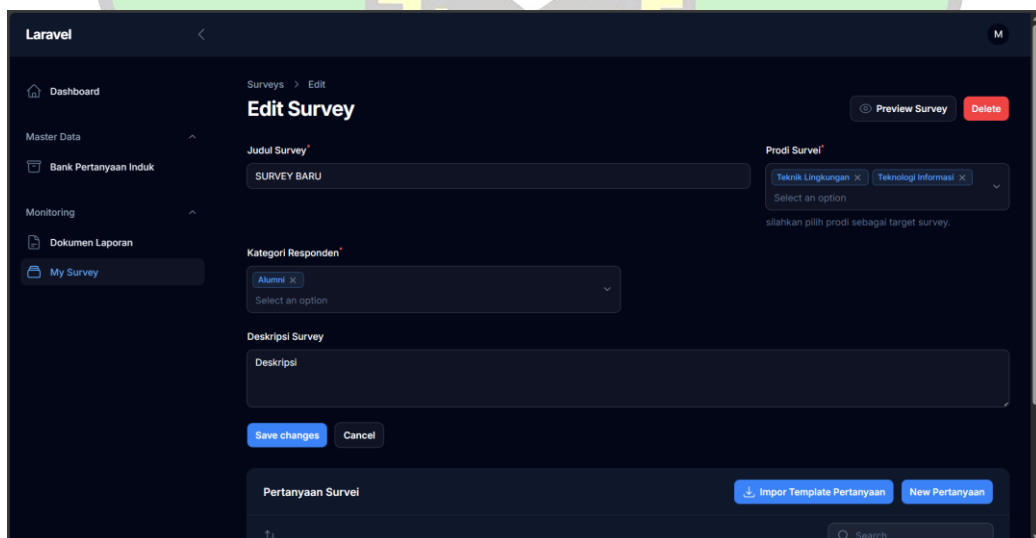


**Gambar 4.3** Halaman *Dashboard Admin*

Gambar 4.3 memperlihatkan tampilan Dashboard Admin. Tampilan ini membuktikan implementasi RBAC di mana akses menu seperti Kelola Akun Admin secara default hanya akan terlihat dan dapat diakses oleh Super Admin, sesuai dengan batasan yang dijelaskan pada *Activity Diagram* Tambah Akun Admin (Gambar 3.7).

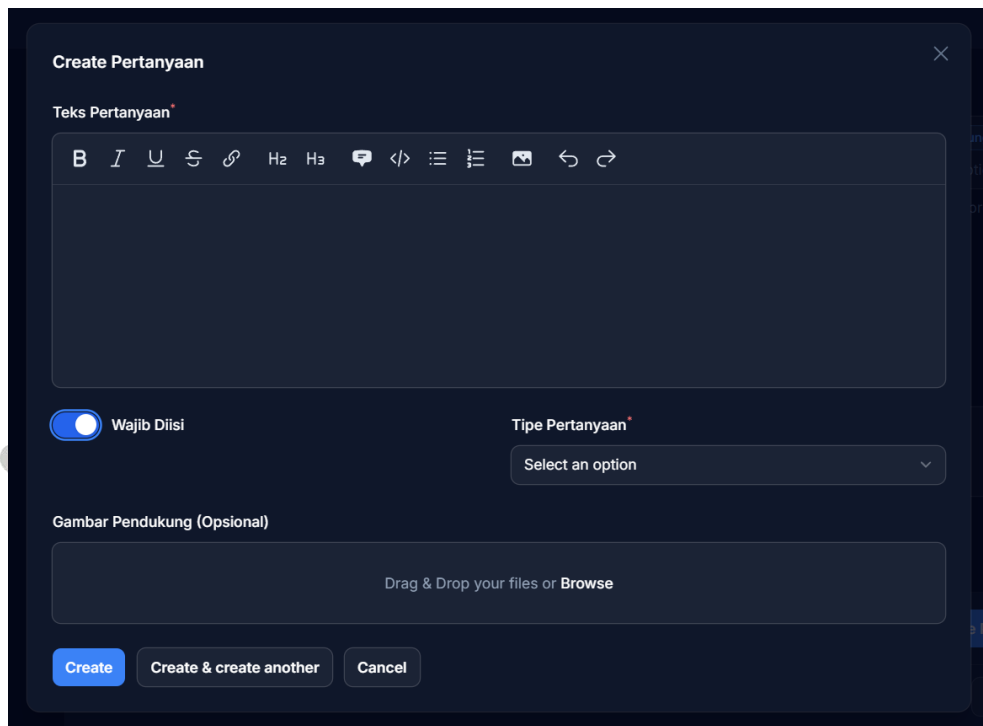
c. Tampilan Kelola Konten Survei

Tampilan ini adalah perwujudan langsung dari *Activity Diagram* Kelola Konten Survei, yang menangani proses kompleks desain instrumen.



**Gambar 4.4** Halaman Form Pembuatan Survei

Gambar 4.4 menunjukkan formulir pembuatan survei baru. Tampilan ini membuktikan implementasi atribut JSON pada *Class Diagram*, terutama pada *field* Target Program Studi. Penggunaan *multi-select* di sini memfasilitasi Admin untuk menargetkan survei ke banyak audiens, sedangkan status awal survei secara otomatis ditetapkan sebagai *DRAFT*.



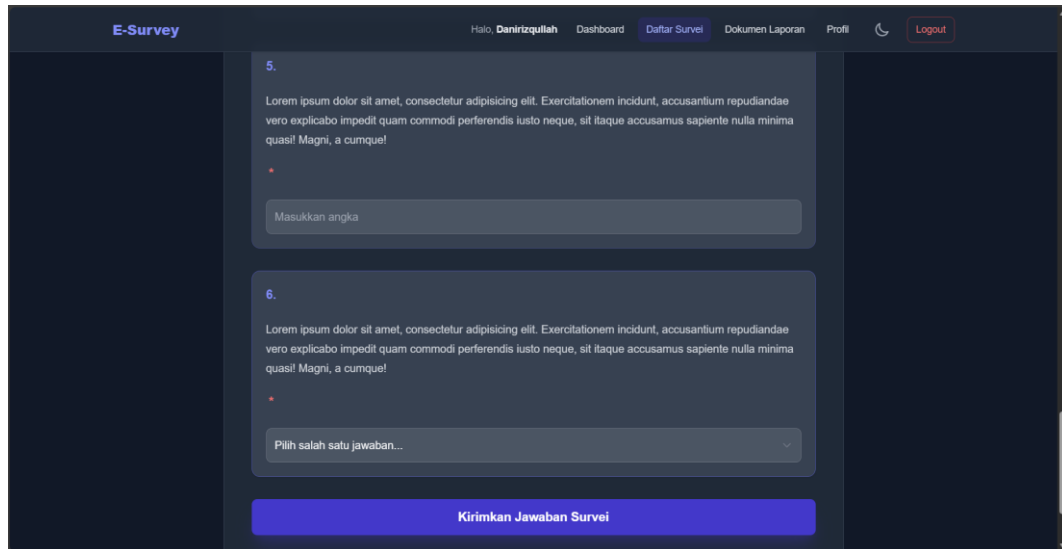
**Gambar 4.5** Halaman tambah pertanyaan

Gambar 4.5 memperlihatkan *Relation Manager* yang digunakan Admin untuk mengelola pertanyaan dan opsi jawaban. Logika krusial yang diimplementasikan di sini adalah:

1. Kunci Skoring: Admin dapat memasukkan Nilai Skor (value *float*) untuk setiap opsi, yang merupakan kunci untuk analisis kuantitatif.
2. Kontrol Integritas: Tampilan ini menerapkan *security gate* di mana tombol "Tambah Baru" dan "Edit" pertanyaan otomatis disembunyikan jika status survei telah berubah menjadi *Published* atau *Closed*, sesuai logika [TIDAK DRAFT] dalam *Activity Diagram* Kelola Konten Survei.

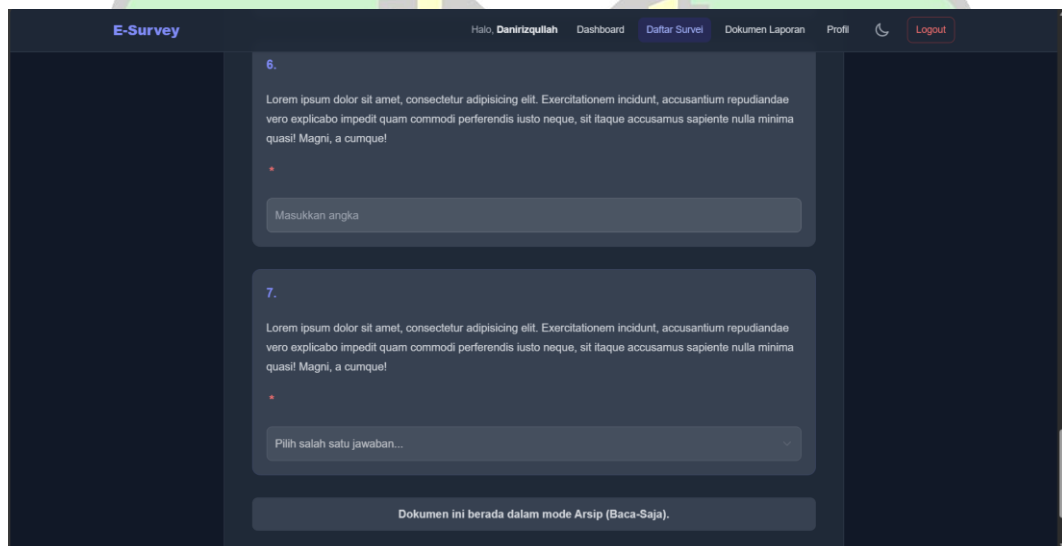
d. Tampilan Pengisian Survei (Responden)

Tampilan ini membuktikan fungsionalitas inti aplikasi sesuai dengan *Activity Diagram* Isi Survei (Gambar 3.4).

The screenshot shows a dark-themed web interface for an 'E-Survey'. At the top, there is a navigation bar with the text 'E-Survey' on the left and 'Halo, Danirizqullah', 'Dashboard', 'Daftar Survei', 'Dokumen Laporan', 'Profil', a moon icon, and a 'Logout' button on the right. The main content area displays two survey questions, numbered 5 and 6. Each question is followed by a red asterisk indicating a required field. Question 5 has a text input field with the placeholder 'Masukkan angka'. Question 6 has a dropdown menu with the placeholder 'Pilih salah satu jawaban...'. At the bottom of the form, there is a prominent blue button labeled 'Kirimkan Jawaban Survei'.

**Gambar 4.6** Halaman pengisian survei responden

Gambar 4.6 menunjukkan tampilan *form* survei yang sedang diisi oleh Responden. Tampilan ini memperlihatkan fungsionalitas validasi wajib diisi (*is\_required*) yang akan memicu pesan kesalahan jika Responden mencoba submit dengan jawaban tidak lengkap.

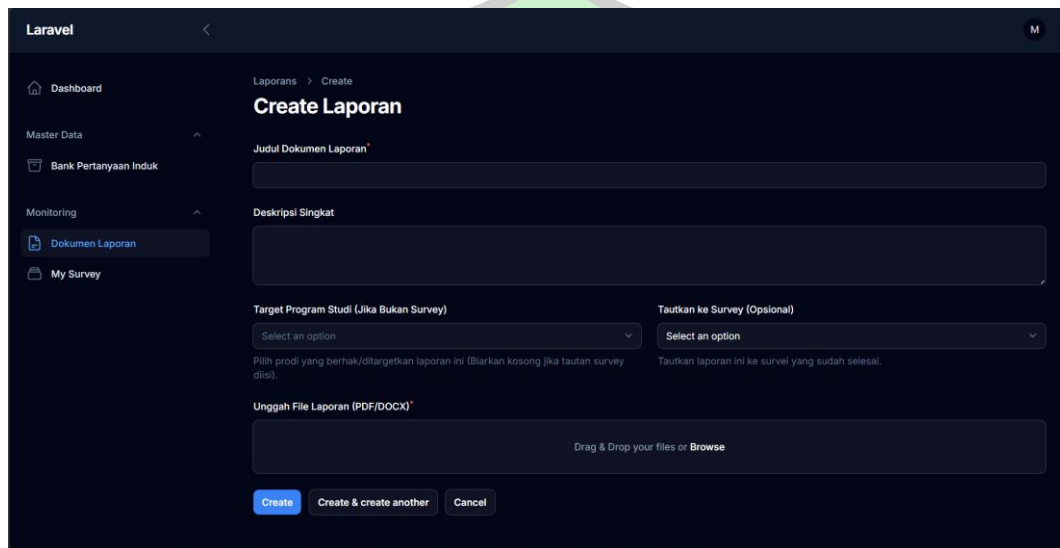
This screenshot shows the same 'E-Survey' application but in a 'Read-only' mode. The navigation bar is identical. The main content area displays two survey questions, numbered 6 and 7, with the same text input and dropdown fields as in the previous image. However, the 'Kirimkan Jawaban Survei' button is replaced by a grey button with the text 'Dokumen ini berada dalam mode Arsip (Baca-Saja)'. The overall appearance is dimmed, indicating that the user cannot interact with the form.

**Gambar 4.7** Halaman pengisian survey mode *Read-only*

Gambar 4.7 membuktikan implementasi *security gate* untuk pencegahan duplikasi dan kontrol status survei. Survei yang telah diisi oleh Responden atau yang berstatus *Closed* akan secara otomatis dialihkan ke tampilan Hanya-Baca, sesuai dengan logika *Activity Diagram* Isi Survei.

e. Tampilan Manajemen Laporan

Tampilan ini mengimplementasikan logika publikasi dokumen laporan sesuai *Activity Diagram* Kelola Konten Publikasi Laporan (Gambar 3.6).

The image shows a web interface for creating a report. On the left is a sidebar with navigation items: Dashboard, Master Data (Bank Pertanyaan Induk), Monitoring (Dokumen Laporan, My Survey). The main content area is titled 'Laporans > Create' and 'Create Laporan'. It contains several form fields: 'Judul Dokumen Laporan\*' (text input), 'Deskripsi Singkat' (text area), 'Target Program Studi (Jika Bukan Survey)' (dropdown menu with 'Select an option'), 'Tautkan ke Survey (Opsional)' (dropdown menu with 'Select an option'), and 'Unggah File Laporan (PDF/DOCX)\*' (file upload area with 'Drag & Drop your files or Browse'). At the bottom are three buttons: 'Create' (highlighted in blue), 'Create & create another', and 'Cancel'.

**Gambar 4.8** Halaman form publikasi laporan

Gambar 4.8 menunjukkan form unggah laporan. Tampilan ini mengimplementasikan field Target Prodi untuk tagging laporan yang relevan, memastikan Responden dapat mencari laporan berdasarkan prodi mereka masing masing.

## 4.2. Pengujian

Tahap pengujian ini dilakukan untuk memverifikasi fungsionalitas sistem yang telah diimplementasikan, memastikan bahwa aplikasi bekerja sesuai dengan kebutuhan yang telah didefinisikan pada tahap Analisis Kebutuhan dan Perancangan Sistem. Pengujian ini menggunakan metode *Black-box Testing*, yang berfokus pada hasil *input* dan *output* sistem tanpa memeriksa struktur internal kode

#### 4.2.1. Pengujian Fungsional

Pengujian ini memverifikasi fungsionalitas inti dan kontrol akses sistem, menguji jalur keberhasilan (*positive testing*) dan kegagalan (*negative testing*). Hasil pengujian dapat dilihat pada Tabel 4.1.

**Tabel 4.1** Pengujian Fungsionalitas Utama (*Black-box Testing*)

No	Fungsi yang Diuji	Skenario Uji	Hasil yang Diharapkan	Hasil Aktual	Keterangan
1	Login dan RBAC	Pengguna Responden berhasil login.	Berhasil, diarahkan ke Dashboard Responden (/responden).	Berhasil	Role Check berjalan.
2	Registrasi Akun	Responden mendaftar dengan email yang sudah terdaftar.	Sistem menolak pendaftaran dan menampilkan pesan email unik.	Berhasil	Validasi Data <i>Input</i> berjalan.
3	Isi Survei	Responden submit dengan pertanyaan wajib kosong.	Sistem menolak submit dan menampilkan pesan kesalahan.	Berhasil	Validasi Jawaban berjalan.
4	Kelola Akun Admin	Super Admin membuat akun Admin dengan Prodi "Lainnya".	Akun dibuat, kolom prodi terisi nama unit kerja dari <i>input</i> teks.	Berhasil	Logika kondisional berjalan.
5	Kelola Laporan	Admin mengunggah file laporan selain format PDF/DOCX.	Sistem menolak unggahan dan menampilkan pesan kesalahan.	Berhasil	Validasi file type berjalan.

Berdasarkan hasil pengujian fungsionalitas pada Tabel 4.1, dapat disimpulkan bahwa aplikasi *e-survey* telah memenuhi kriteria keberhasilan sesuai dengan rancangan *Activity Diagram* yang telah ditetapkan. Pengujian ini menjawab

rumusan masalah mengenai akurasi dan kemudahan manajemen sistem melalui poin analisis berikut:

1. **Akurasi Validasi Data:** Pada pengujian nomor 2, 3, dan 5, sistem menunjukkan tingkat akurasi tinggi dalam mendeteksi data tidak valid seperti duplikasi email dan format file yang salah. Ini membuktikan bahwa *Laravel 12* mampu menjaga integritas data sebelum masuk ke basis data.
2. **Keamanan dan Hak Akses (RBAC):** Pengujian nomor 1 dan 4 membuktikan mekanisme *Role-Based Access Control* berjalan sempurna dalam membedakan hak akses antara Responden, Admin, dan Super Admin secara otomatis.
3. **Kemudahan Pengelolaan:** Sistem terbukti mampu mengotomatisasi proses administratif yang sebelumnya dilakukan secara manual (seperti validasi format file dan pengecekan kelengkapan data), sehingga meminimalkan risiko kesalahan manusia (*human error*).

Secara keseluruhan, hasil pengujian fungsional ini memberikan bukti teknis bahwa sistem *e-survey* ini handal untuk digunakan sebagai platform terpusat di Fakultas Sains dan Teknologi.

#### 4.2.2. Pengujian Integritas Data dan Keamanan Kritis

Pengujian ini secara khusus memverifikasi *security gate* dan logika bisnis yang kompleks yang dirancang untuk menjaga integritas data, sesuai dengan implementasi di *SurveyController* dan *SurveyResource*. Hasil pengujian ini dapat dilihat pada table 4.2

**Tabel 4.2** Pengujian Integritas Data dan Keamanan Kritis

No	Logika yang Diuji	Skenario Uji	Hasil yang Diharapkan (Sesuai Kode & Diagram)	Hasil Aktual	Keterangan
1	Kontrol Status Edit	Admin mencoba mengedit pertanyaan pada	Aksi ditolak (tombol Edit disembunyikan), sesuai <i>Activity Diagram</i> .	Berhasil	Soal survei tetap aman.

		survei berstatus Published.			
2	Pencegahan Duplikasi	Responden mencoba submit survei yang sudah pernah diisi sebelumnya.	Sistem menolak submit dan mengarahkan ke halaman survei format <i>read-only</i> .	Berhasil	Tidak ada data ganda.
3	Pengujian Skoring Data	Responden mengisi skor 5, 4, 3; sistem menghitung rata-rata skor di laporan.	Rata-rata skor (Mean) dihitung dengan benar dari nilai <i>float</i> di tabel <i>options</i> .	Berhasil	Perhitungan skor akurat.
4	Targeting Data	Responden Prodi X mengakses menu Laporan.	Hanya laporan dengan <i>target_prodi</i> mencakup Prodi X yang ditampilkan.	Berhasil	Logika filter berjalan.

Berdasarkan hasil pengujian pada Tabel 4.2, sistem telah terbukti mampu menjalankan aturan main yang cukup rumit untuk menjaga keaslian data survei. Pengujian ini memberikan gambaran mengenai ketepatan dan keamanan sistem melalui poin-poin berikut:

- **Keamanan Soal Survei:** Pengujian nomor 1 membuktikan bahwa sistem memiliki pengamanan yang ketat. Dengan mengunci survei yang sudah disebar (*Published*), sistem menjamin soal tidak bisa diubah-ubah saat orang sedang mengisi. Hal ini sangat penting agar data tetap konsisten dan tidak berantakan karena adanya perubahan pertanyaan secara tiba-tiba.
- **Keaslian Data Responden:** Pada pengujian nomor 2, fitur cek pengisian berhasil mencegah orang mengisi berkali-kali. Ini mengatasi kelemahan platform survei biasa di mana satu orang bisa memberikan banyak jawaban yang bisa merusak hasil akhir. Dengan cara ini, sistem menjamin satu akun hanya bisa memberikan satu kali jawaban untuk setiap survei.
- **Ketepatan Perhitungan Angka:** Pengujian nomor 3 membuktikan bahwa sistem sangat akurat dalam menghitung nilai. Sistem bisa langsung

mengolah angka desimal dari basis data secara otomatis untuk menghasilkan nilai rata-rata (*Mean*) yang tepat. Ini membuktikan bahwa sistem sangat membantu dalam menyajikan laporan tanpa perlu lagi menghitung manual di Excel.

- **Laporan Tepat Sasaran:** Pengujian nomor 4 menunjukkan bahwa aturan pembatasan prodi berjalan dengan pas. Hal ini memastikan kerahasiaan data terjaga, di mana laporan hanya bisa dilihat oleh pihak yang memang diberikan izin oleh Admin.

Secara keseluruhan, pengujian ini mempertegas bahwa aplikasi *e-survey* yang dikembangkan memiliki keunggulan dalam hal kontrol data dibandingkan cara lama atau platform gratisan. Keberhasilan ini membuktikan bahwa sistem tidak hanya jalan secara teknis, tapi juga punya logika koding yang kuat untuk mendukung pengambilan keputusan yang benar di Fakultas Sains dan Teknologi.

#### **4.3. Pembahasan Hasil**

Bagian ini membahas hasil pembuatan dan pengujian sistem yang sudah dijelaskan sebelumnya. Pembahasan ini difokuskan untuk melihat sejauh mana aplikasi berhasil menyelesaikan masalah pengelolaan survei di Fakultas Sains dan Teknologi.

##### **4.3.1. Analisis Implementasi Fitur Kritis**

Analisis ini melihat bagaimana fitur-fitur teknis yang dibangun berhasil menjaga keamanan data dan merapikan administrasi survei:

1. **Keamanan dan Ketepatan Data:** Berdasarkan hasil uji coba, sistem terbukti dapat menyimpan data secara aman. Penggunaan logika koding yang tepat memastikan jawaban responden tersimpan utuh tanpa ada risiko data yang hilang. Selain itu, sistem sukses mengubah pilihan jawaban menjadi angka laporan statistik yang benar.
2. **Kunci Status Survei:** Pengamanan yang mencegah perubahan soal pada survei yang sudah disebar (*Published*) telah berjalan dengan baik. Hal ini menjamin bahwa data yang dikumpulkan tetap sama dan tidak berubah di tengah jalan, sehingga mengurangi risiko kesalahan data.

3. **Pengaturan Izin Akses:** Pembatasan hak akses sesuai peran pengguna berhasil memisahkan wewenang antara admin fakultas dan program studi. Hal ini memastikan keamanan data dan kejelasan tanggung jawab masing-masing unit kerja.

#### 4.3.2. Pembahasan Pencapaian Tujuan Penelitian

Hasil pembuatan dan pengujian secara keseluruhan membuktikan bahwa tujuan penelitian ini telah tercapai, dengan rincian sebagai berikut:

1. **Berhasil Membangun Sistem Terpusat:** Aplikasi *e-survey* berbasis *Laravel* 12 telah sukses dibangun dan berfungsi sebagai wadah tunggal bagi seluruh kegiatan survei di Fakultas Sains dan Teknologi. Sistem ini berhasil menyelesaikan masalah data yang sebelumnya tercecer di banyak tempat dengan menyatukan semuanya ke dalam satu basis data terpusat.
2. **Berhasil Melewati Pengujian:** Seluruh fitur utama aplikasi, mulai dari pengelolaan bank soal hingga pembuatan laporan otomatis, telah lulus pengujian menggunakan metode *Black-box Testing*. Sistem terbukti mampu menampilkan laporan hasil survei per prodi tanpa perlu lagi dihitung manual, yang merupakan solusi atas kendala pada proses lama

Secara keseluruhan, sistem *e-survey* ini terbukti berjalan dengan baik dan mampu menyediakan solusi digital yang rapi untuk pengelolaan survei di Fakultas Sains dan Teknologi.



## BAB V

### KESIMPULAN

#### 5.1 Kesimpulan

Berdasarkan hasil rancang bangun dan pengujian sistem Website *e-survey* Fakultas Sains dan Teknologi menggunakan metode *Waterfall* dan framework *Laravel* 12, maka dapat ditarik beberapa kesimpulan utama sebagai berikut:

1. Penelitian ini telah berhasil merancang dan membangun aplikasi *e-survey* berbasis web menggunakan framework *Laravel* 12 yang mampu menyatukan proses pengelolaan kuesioner, data responden, dan pelaporan hasil survei ke dalam satu sistem yang terpusat di Fakultas Sains dan Teknologi. Penggunaan struktur *Model-View-Controller* (MVC) dan platform admin *Filament* dalam pengembangan ini terbukti dapat mengelola data kuesioner dan instrumen penelitian menjadi lebih terorganisir dibandingkan proses manual sebelumnya.
2. Hasil pengujian fungsionalitas menggunakan metode *Black-box Testing* menunjukkan bahwa seluruh fitur aplikasi telah berjalan dengan benar sesuai dengan kebutuhan perancangan yang ditetapkan. Sistem telah terbukti mampu melakukan validasi pada setiap masukan data untuk mencegah adanya pengisian kuesioner ganda oleh responden yang sama, serta dapat mengolah hasil jawaban tersebut menjadi laporan statistik secara otomatis.

## DAFTAR PUSTAKA

- Ananda, A. T., Butsianto, S., & Sulaeman, A. A. (2024). Penerapan Metode *Waterfall* pada Sistem Informasi Inventory Berbasis Website. *Journal of Information System Research (JOSH)*, 5(4), 1147–1155.  
<https://doi.org/10.47065/josh.v5i4.5669>
- Apiag, C., Cadiz, E., & Lincopinis, D. (2023). *A Review on PHP Programming Language*.
- Ardi, Y. M., Warsino, & Afghani, F. U. (2023). Perancangan Program Survey Online (Surveyasia) Berbasis Web Pada PT. Citiasia Inc Jakarta Selatan. *Jurnal Teknik AMATA*, 4(1), 27–31.  
<https://doi.org/10.55334/jtam.v4i1.86>
- Back4App. (2023). *Frontend - Laravel 12.x - The PHP Framework For Web Artisans*. <https://Laravel.com/docs/12.x/frontend>
- Cristian Richardo Anin, Nia Ambarsari, & Sinung Suakanto. (2023). Development of a Web-Based *E-survey* Application for Assessing Good Corporate Governance Understanding at PT. XYZ. *Digital Zone: Jurnal Teknologi Informasi Dan Komunikasi*, 14(2), 115–127.  
<https://doi.org/10.31849/digitalzone.v14i2.14792>
- Endra, R. Y., Aprilinda, Y., Dharmawan, Y. Y., & Ramadhan, W. (2021). Analisis Perbandingan Bahasa Pemrograman PHP *Laravel* dengan PHP Native pada Pengembangan Website. *EXPERT: Jurnal Manajemen Sistem Informasi Dan Teknologi*, 11(1), 48.  
<https://doi.org/10.36448/expert.v11i1.2012>
- Fadilillah, F., Amir, F., Riyanto, R., Hasibuan, R. S., & Maryam, S. I. (2025). Sistem Informasi Survei Kepuasan Masyarakat dengan Framework *Laravel* pada Pelayanan Publik Pemerintah Daerah. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 5(3).  
<https://doi.org/10.57152/malcom.v5i3.1772>

- Fitra, M. F., & Syukhri, S. (2023). Rancang Bangun *E-survey* Indeks Kepuasan Masyarakat Terhadap Pelayanan Pemerintahan Desa. *Voteteknika (Vocational Teknik Elektronika Dan Informatika)*, 11(1), 80. <https://doi.org/10.24036/voteteknika.v11i1.120587>
- Gibran, C., Rafika Dewi, A., & Hadinata, E. (2024). Implementasi Framework *Laravel* Untuk Pengembangan Website Penjualan Ayam Potong Dengan Pemanfaatan Midtrans Menggunakan Metode Fast. *Jurnal Ilmu Komputer Dan Sistem Informasi (JIKOMSI)*, 7(1), 246–253. <https://doi.org/10.55338/jikomsi.v7i1.2920>
- Husbi, F. R., & Ichwani, A. (2024). Digitalisasi Proses Survei Badan Litbang dan Diklat Kementerian Agama Republik Indonesia Melalui Pembuatan Aplikasi *E-survey* Berbasis Web. *METHOMIKA Jurnal Manajemen Informatika Dan Komputerisasi Akuntansi*, 8(2), 222–233. <https://doi.org/10.46880/jmika.Vol8No2.pp222-233>
- Kurniawan, V., Amroni, & Yose, I. (2023). Perancangan Sistem Informasi Administrasi Kependudukan Pada Desa Seri Sembilan. *Jurnal Manajemen Teknologi Dan Sistem Informasi (JMS)*, 3(2), 542–553. <https://doi.org/10.33998/jms.2023.3.2.1416>
- Laravel*. (2024). *Top 10 Frontend & Backend Frameworks*. <https://blog.back4app.com/front-end-and-back-end-frameworks/>
- Nuril Jazilah, A., Prasetyo, J. A., Ratri, I. N., Subono, S., & Hakim, L. (2024a). Rancang bangun sistem survei kepuasan pemangku kepentingan berbasis web di Politeknik Negeri Banyuwangi. *JURNAL ELTEK*, 22(2), 50–65. <https://doi.org/10.33795/eltek.v22i2.5429>
- Nuril Jazilah, A., Prasetyo, J. A., Ratri, I. N., Subono, S., & Hakim, L. (2024b). Rancang bangun sistem survei kepuasan pemangku kepentingan berbasis web di Politeknik Negeri Banyuwangi. *JURNAL ELTEK*, 22(2), 50–65. <https://doi.org/10.33795/eltek.v22i2.5429>
- Patmanthara, S., Perdana Putra, M. R., & Soraya, D. U. (2024). Analysis and Design of Responsive Web-Based Learning Media Equipped with

- Pomodoro Feature in the Course of Software Engineering. *Jurnal Ilmu Pendidikan*, 30(1), 8. <https://doi.org/10.17977/um048v30i1p7-18>
- Perdana, A., Farhana, N. A., Harliana, P., & Karo Karo, I. M. (2024). Web-Based Application Development using PHP-Native Framework on Agent of Change Integrity Zone Information System. *Sinkron*, 8(4), 2458–2468. <https://doi.org/10.33395/sinkron.v8i4.14118>
- Permana, K. E., Sophan, M. K., Muntasa, A., & Rahmat, A. B. (2023). PERBANDINGAN KINERJA QUERY SQL JOIN TABLES DENGAN MENGGUNAKAN INDEX. *Jurnal Simantec*, 11(2), 241–249. <https://doi.org/10.21107/simantec.v11i2.20288>
- Raynaldi, J., & Somya, R. (2023). PERANCANGAN APLIKASI E-KANTIN BERBASIS WEBSITE MENGGUNAKAN FRAMEWORK LARAVEL. *Jurnal Pendidikan Teknologi Informasi (JUKANTI)*, 6(2), 356–366. <https://doi.org/10.37792/jukanti.v6i2.1052>
- Saputri, N., Safriadi, S., Dahlia, Z., & Rahmadani, R. (2024). Optimizing Web-Based Survey Applications with *Laravel* and Cloud Computing. *Journal of Artificial Intelligence and Engineering Applications (JAIEA)*, 4(1), 557–561. <https://doi.org/10.59934/jaiea.v4i1.691>
- Srikarthik, K., Singh Jebaraj, B., Bhavani, K., Lahari, K. L., Lakshmi Prasanna, B., & K, V. (2025). Survey App: Rating and Feedback System Application. *2025 5th International Conference on Expert Clouds and Applications (ICOECA)*, 628–631. <https://doi.org/10.1109/ICOECA66273.2025.00114>
- Suarsana, I. K. S., I Gede Juliana Eka Putra, & A.A. Istri Ita Paramitha. (2023). RANCANG BANGUN SISTEM PENGUKURAN INDEKS KEPUASAN MASYARAKAT (IKM) BERBASIS WEB DI PEMERINTAH KABUPATEN BADUNG. *Jurnal Informatika Teknologi Dan Sains (Jinteks)*, 5(4), 618–625. <https://doi.org/10.51401/jinteks.v5i4.3385>

Syahira, A. S., & Muhamad Bahrul Ulum. (2024). Rancang Bangun Aplikasi Indeks Kepuasan Pasien RS Qadr dengan Integrasi Survei Online Berbasis Website. *Jurnal Komputasi*, 12(1), 79–80.

<https://doi.org/10.23960/komputasi.v12i1.253>

Syaifudin, Y. W., Fatmawati, T., Udjir, A. D., Saputra, P. Y., Funabiki, N., Mentari, M., Setiadi, C., & Rahmadani, A. A. (2025). Implementation of Test-Driven Approach to Empower Self-Learning in PHP Web Programming Practice. *International Journal of Computing*, 318–327.

<https://doi.org/10.47839/ijc.24.2.4015>

Utami, A. P., Erzed, N., Uddin, B., & Sekti, B. A. (2025). Perancangan Aplikasi Website Survei Minat Mahasiswa Terhadap Organisasi Kemahasiswaan di Universitas Esa Unggul Kampus Bekasi.

*INNOVATIVE: Journal Of Social Science Research*, 5(3), 160–175.

Wikipedia contributors. (2025). *PHP - Wikipedia*.

[https://en.wikipedia.org/wiki/PHP?utm\\_source=chatgpt.com](https://en.wikipedia.org/wiki/PHP?utm_source=chatgpt.com)

Wulansari, A., & Kristiana, T. (2022). Sistem Informasi Survey Online Berbasis Website Pada Pt. Citi Asia International Menggunakan Metode *Waterfall*. *Simpatik: Jurnal Sistem Informasi Dan Informatika*, 2(2), 82–

87. <https://doi.org/10.31294/simpatik.v2i2.1293>

جامعة الرانيري

A R - R A N I R Y

## LAMPIRAN

### 1. Migrasi

#### 1.1. Kode migrasi user

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('users', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('email')->unique();
            $table->timestamp('email_verified_at')->nullable();
            $table->string('password');
            $table->enum('role', ['superadmin', 'admin', 'responden'])-
            >default('responden');
            $table->string('program_studi')->nullable();
            $table->rememberToken();
            $table->timestamps();
        });

        Schema::create('password_reset_tokens', function (Blueprint
        $table) {
            $table->string('email')->primary();
            $table->string('token');
            $table->timestamp('created_at')->nullable();
        });

        Schema::create('sessions', function (Blueprint $table) {
            $table->string('id')->primary();
            $table->foreignId('user_id')->nullable()->index();
            $table->string('ip_address', 45)->nullable();
            $table->text('user_agent')->nullable();
            $table->longText('payload');
            $table->integer('last_activity')->index();
        });
    }
    public function down(): void
    {
        Schema::dropIfExists('users');
        Schema::dropIfExists('password_reset_tokens');
        Schema::dropIfExists('sessions');
    }
};
```

## 1.2. Kode migrasi survey

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('surveys', function (Blueprint $table) {
            $table->id();
            $table->foreignId('created_by')->constrained('users')->onDelete('cascade');
            $table->string('title');
            $table->json('categories')->nullable();
            $table->json('program_studi')->nullable();
            $table->text('description')->nullable();
            $table->date('start_date')->nullable();
            $table->date('end_date')->nullable();
            $table->enum('status', ['draft', 'published', 'Closed'])->default('draft');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('surveys');
    }
};
```

## 1.3. Kode migrasi questions

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('questions', function (Blueprint $table) {
```

```

        $table->id();
        $table->foreignId('survey_id')->constrained()-
>onDelete('cascade');
        $table->text('question_text');
        $table->boolean('is_required')->default(false);
        $table->string('image')->nullable();
        $table->enum('question_type', ['text', 'radio', 'checkbox',
'textarea', 'number', 'select']->default('text');
        $table->integer('order')->default(0);

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

/**
 * Reverse the migrations.
 */
public function down(): void
{
    Schema::dropIfExists('questions');
}
};

```

#### 1.4. Kode migrasi options

```

<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        // Pastikan tabel options dibuat setelah tabel questions
        Schema::create('options', function (Blueprint $table) {
            $table->id();
            $table->foreignId('question_id')->constrained()-
>onDelete('cascade');
            $table->string('option_text');
            // Kolom KRITIS: Nilai numerik (skor) untuk analisis
            $table->float('value')->default(0);
            // Kolom KRITIS: Urutan tampilan opsi
            $table->integer('order')->default(0);
            $table->timestamps();
        });
    }
    public function down(): void
    {
        Schema::dropIfExists('options');
    }
};

```

## 1.5. Kode migrasi responses

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('responses', function (Blueprint $table) {
            $table->id();
            $table->foreignId('user_id')->constrained()-
            >onDelete('cascade');
            $table->foreignId('survey_id')->constrained()-
            >onDelete('cascade');
            $table->timestamp('submitted_at')->useCurrent();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('responses');
    }
};
```

## 1.6. Kode migrasi Answers

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('Answers', function (Blueprint $table) {
            $table->id();
            $table->foreignId('response_id')->constrained()-
            >onDelete('cascade');
            $table->foreignId('question_id')->constrained()-
            >onDelete('cascade');
```

```

                $table->foreignId('option_id')->nullable()-
>constrained('options')->onDelete('cascade');
                $table->text('Answer_text')->nullable();
                $table->timestamps();
            });
        }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('Answers');
    }
};

```

### 1.7. Kode migrasi instrument question

```

<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('instrument_questions', function (Blueprint $table)
        {
            $table->id();
            $table->foreignId('created_by')->constrained('users')->
>onDelete('cascade');
            $table->string('template_group')->index();
            $table->text('question_text');
            $table->enum('question_type', ['text', 'radio', 'checkbox',
'textarea', 'number', 'select']->default('text');
            $table->boolean('is_required')->default(false);
            $table->string('image')->nullable();
            $table->json('options')->nullable();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('instrument_questions');
    }
};

```

## 1.8. Kode migrasi laporans

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('laporans', function (Blueprint $table) {
            $table->id();
            $table->foreignId('created_by')->constrained('users')->onDelete('cascade');
            $table->string('title')->unique();
            $table->string('survey_relation')->nullable();
            $table->json('target_prodi')->nullable();
            $table->text('description')->nullable();
            $table->string('file_path');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('laporans');
    }
};
```

## 2. Model

### 2.1. Model user

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;

class User extends Authenticatable
{
    use HasFactory, Notifiable;

    /**
     * The attributes that are mass assignable.
     */
```

```

protected $fillable = [
    'name',
    'email',
    'password',
    'role',
    'program_studi',
];

protected $hidden = [
    'password',
    'remember_token',
];

protected function casts(): array
{
    return [
        'email_verified_at' => 'datetime',
        'password' => 'hashed',
    ];
}

public function createdSurveys()
{
    return $this->hasMany(Survey::class, 'created_by');
}

public function responses()
{
    return $this->hasMany(Response::class);
}
}

```

## 2.2. Model Survey

```

<?php
namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Support\Carbon;

class Survey extends Model
{
    use HasFactory;

    protected $fillable = [
        'title',
        'categories',
        'description',
        'program_studi',
        'start_date',
        'end_date',
        'status',
        'created_by',
    ];
}

```

```

];

protected $casts = [
    'start_date' => 'date',
    'end_date' => 'date',
    'categories' => 'array',
    'program_studi' => 'array',
];

public function creator()
{
    return $this->belongsTo(User::class, 'created_by');
}

public function questions()
{
    return $this->hasMany(Question::class);
}

public function responses()
{
    return $this->hasMany(Response::class);
}

protected static function booted()
{
    static::retrieved(function ($survey) {
        if (
            $survey->status === 'published'
            && $survey->end_date
            && Carbon::now()->greaterThan($survey->end_date)
        ) {
            $survey->update(['status' => 'Closed']);
        }
    });
}
}
}

```

### 2.3. Model Response

```

<?php

namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Response extends Model
{
    use HasFactory;

    protected $fillable = ['user_id', 'survey_id', 'submitted_at'];

    public function user()
    {
        return $this->belongsTo(User::class);
    }
}

```

```

public function survey()
{
    return $this->belongsTo(Survey::class);
}

public function Answers()
{
    return $this->hasMany(Answer::class);
}
}

```

## 2.4. Model Question

```

<?php
namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\HasMany;
use Illuminate\Database\Eloquent\Relations\BelongsTo;
use Illuminate\Database\Eloquent\Builder;

class Question extends Model
{
    use HasFactory;

    protected $fillable = [
        'survey_id',
        'question_text',
        'question_type',
        'is_required',
        'image',
        'order',
    ];

    protected $casts = [
        'is_required' => 'boolean',
    ];

    public function survey(): BelongsTo
    {
        return $this->belongsTo(Survey::class);
    }

    public function options(): HasMany
    {
        return $this->hasMany(Option::class)->orderBy('order');
    }

    public function Answers(): HasMany
    {
        return $this->hasMany(Answer::class);
    }
}

```

```

protected static function booted()
{
    static::addGlobalScope('orderByOrder', function (Builder $builder)
    {
        $builder->orderBy('order', 'asc');
    });
}
}

```

## 2.5. Mode Options

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\BelongsTo;

class Option extends Model
{
    use HasFactory;

    protected $fillable = [
        'question_id',
        'option_text',
        'value',
        'order',
    ];

    protected $casts = [
        'value' => 'float',
        'order' => 'integer',
    ];

    public function question(): BelongsTo
    {
        return $this->belongsTo(Question::class);
    }
}

```

## 2.6. Model Laporan

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\BelongsTo;
use Illuminate\Support\Facades\Auth;

class Laporan extends Model
{
    use HasFactory;
}

```

```

protected $table = 'laporans';

protected $fillable = [
    'title',
    'survey_relation',
    'description',
    'file_path',
    'created_by',
    'target_prodi',
];

protected $casts = [
    'target_prodi' => 'array',
];

public function creator(): BelongsTo
{
    return $this->belongsTo(User::class, 'created_by');
}

protected static function boot()
{
    parent::boot();

    static::creating(function ($model) {
        if (Auth::check()) {
            $model->created_by = Auth::id();
        }
    });
}
}

```

## 2.7. Model Answer

```

<?php
namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\BelongsTo;

class Answer extends Model
{
    use HasFactory;

    protected $fillable = [
        'response_id',
        'question_id',
        'Answer_text',
        'option_id',
    ];

    public function response(): BelongsTo
    {
        return $this->belongsTo(Response::class);
    }
}

```

```

    }

    public function question(): BelongsTo
    {
        return $this->belongsTo(Question::class);
    }

    public function option(): BelongsTo
    {
        return $this->belongsTo(Option::class, 'option_id');
    }
}

```

## 2.8. Model InstrumentQuestion

```

<?php
namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\BelongsTo;
use Illuminate\Support\Facades\Auth;

class InstrumentQuestion extends Model
{
    use HasFactory;

    protected $fillable = [
        'template_group',
        'question_text',
        'question_type',
        'is_required',
        'image',
        'options',
    ];

    protected $casts = [
        'is_required' => 'boolean',
        'options' => 'array',
    ];
    protected static function boot()
    {
        parent::boot();

        static::creating(function ($model) {
            if (Auth::check()) {
                $model->created_by = Auth::id();
            }
        });
    }

    public function creator(): BelongsTo
    {
        return $this->belongsTo(User::class, 'created_by');
    }
}

```

```
}  
}
```

### 3. Middleware

#### 3.1. Checkrole.php

```
<?php  
  
namespace App\Http\Middleware;  
  
use Closure;  
use Illuminate\Http\Request;  
use Illuminate\Support\Facades\Auth;  
use Symfony\Component\HttpFoundation\Response;  
  
class CheckRole  
{  
    /**  
     * Handle an incoming request.  
     *  
     * @param \Illuminate\Http\Request $request  
     * @param \Closure(\Illuminate\Http\Request):  
     * (\Symfony\Component\HttpFoundation\Response) $next  
     * @param mixed ...$roles  
     */  
    public function handle(Request $request, Closure $next, ...$roles):  
Response  
    {  
        if (!Auth::check()) {  
            return redirect()->route('login');  
        }  
  
        if (!in_array(Auth::user()->role, $roles)) {  
            abort(403, 'Akses ditolak.');        }  
  
        return $next($request);  
    }  
}
```

#### 4. Logika Pengolahan Statistik (ControllerSurveyReport.php)

##### 4.1. Perhitungan Frekuensi dan Skor

```
foreach ($options as $option) {
    $count = $AnsweredOptionIds->filter(fn($id) => (int) $id === (int)
    $option->id)->count();
    $chartData[] = ['label' => (string) $optText, 'value' => (int) $count];

    // Rumus: Frekuensi * Bobot Nilai Opsi
    $totalScore += $count * ($option->value ?? 0);
}
```

##### 4.2. Kalkulasi Nilai Rata-rata (Mean)

```
$averageScore = $totalResponses > 0 ? round($totalScore / $totalResponses,
2) : 0;

$questionResult['stats'] = [
    'total_responses' => $totalResponses,
    'average_score' => $averageScore,
];
```

##### 4.3. Visualisasi Data Dinamis (Chart Generation)

```
// Mengonversi data statistik menjadi objek grafik (Bar/Pie)
$chartConfig = [
    "type" => $type === 'checkbox' ? 'pie' : 'bar',
    "data" => [
        "labels" => collect($chartData)->pluck('label')->toArray(),
        "datasets" => [[
            "data" => collect($chartData)->pluck('value')->toArray(),
            "backgroundColor" => $bg->toArray(),
        ]]
    ]
];
$url = "https://quickchart.io/chart?c="
urlencode(json_encode($chartConfig));
```

##### 4.4. Rendering PDF dengan DomPDF

```
public function exportPdf(Survey $survey) {
    $results = $this->prepareResults($survey, true); // Flag true untuk
generate chart base64

    Pdf::setOptions(['isRemoteEnabled' => true]); // Kritis untuk memuat
grafik dari API
    $pdf = Pdf::loadView('reports.report-pdf', compact('survey',
'results'))
        ->setPaper('a4', 'portrait');

    return $pdf->download('laporan-survey.pdf');
}
```

#### 5. Logika Pengisian dan Validasi Survei (SurveyController.php)

### 5.1. Manajemen Daftar Survei Berdasarkan Status Responden

```
// Mendapatkan ID survei yang sudah diisi oleh responden ini
$completedSurveyIds = $user->responses()->pluck('survey_id')->unique();

// Query Survei Aktif: Belum diisi dan status Published
$activeSurveys = Survey::where('status', 'published')
    ->whereNotIn('id', $completedSurveyIds)
    ->latest()->get();
```

### 5.2. Pembangunan Aturan Validasi Otomatis (*Dynamic Validation*)

```
protected function buildValidationRules($questions) {
    $rules = [];
    foreach ($questions as $question) {
        $fieldName = 'Answers.' . $question->id;
        if ($question->is_required) $rules[$fieldName][] = 'required';
        if ($question->question_type === 'number') $rules[$fieldName][] =
'numeric';

        // Validasi Integritas: Pastikan pilihan jawaban ada di database
        if (in_array($question->question_type, ['radio', 'select',
'checkbox'])) {
            $validOptions = $question->options->pluck('id')->toArray();
            $rules[$fieldName][] = Rule::in($validOptions);
        }
    }
    return $rules;
}
```

### 5.3. Penyimpanan Massal Jawaban (*Batch Insert & Transaction*)

```
DB::beginTransaction();
try {
    $response = Response::create(['user_id' => $userId, 'survey_id' =>
$survey->id]);

    // Proses pengumpulan jawaban ke dalam array untuk Batch Insert
    foreach ($questions as $question) {
        $AnswersToStore[] = [
            'response_id' => $response->id,
            'question_id' => $question->id,
            'Answer_text' => $AnswerValue,
        ];
    }
    Answer::insert($AnswersToStore); // Simpan semua jawaban dalam satu
query
    DB::commit();
} catch (\Exception $e) { DB::rollBack(); }
```

## 6. Logika Manajemen Survei dan Automasi Admin

### 6.1. Mekanisme Publikasi dan Penjadwalan Survei

```
Action::make('togglePublish')
->action(function (array $data, Survey $record) {
    if ($record->status === 'published') {
        $record->update(['status' => 'draft']); // Tarik kembali
survei
    } else {
        $record->update([
            'status' => 'published',
            'start_date' => now(), // Set otomatis waktu mulai
            'end_date' => $data['end_date'],
        ]);
    }
})
->form([
    DatePicker::make('end_date')->required()->minDate(now()), //
Validasi tanggal masa depan
]);
```

### 6.2. Automasi Duplikasi Survei Beserta Relasi (*Deep Copy*)

```
Action::make('duplicate')
->action(function ($record) {
    $newSurvey = $record->replicate(); // Kloning header survei
    $newSurvey->fill(['status' => 'draft', 'created_by' =>
Auth::id()])->save();

    foreach ($record->questions()->with('options')->get() as
$question) {
        $newQuestion = $question->replicate();
        $newQuestion->survey_id = $newSurvey->id; // Hubungkan ke
survei baru
        $newQuestion->save();

        foreach ($question->options as $option) {
            $newOption = $option->replicate();
            $newOption->question_id = $newQuestion->id;
            $newOption->save();
        }
    }
});
```

### 6.3. Isolasi Data Berbasis Pengguna (*Data Isolation*)

```
public static function getEloquentQuery(): Builder
{
    // Filter otomatis: Admin hanya melihat data yang mereka buat
    return parent::getEloquentQuery()->where('created_by', Auth::id());
}
```