BAB III METODE PENELITIAN

3.1 Lokasi dan Waktu Penelitian

1. Lokasi Penelitian

Penelitian ini dilaksanakan pada Divisi Compliance and Risk Management (CRM) di PT BRI Insurance (BRINS), yang menjadi objek studi dalam perancangan dan pengujian sistem Risk Register berbasis web.

2. Waktu Penelitian

Penelitian dilaksanakan selama lima bulan, yaitu dari September 2025 hingga Januari 2026. Rentang waktu tersebut mencakup tahapan analisis kebutuhan, perancangan sistem, implementasi, pengujian menggunakan metode Black Box Testing dan User Acceptance Testing (UAT), hingga penyusunan laporan penelitian.

3.2 Metode Pengumpulan Data

1. **Observasi**

   Observasi dilakukan dengan cara mengamati secara langsung proses pencatatan dan pengelolaan risiko yang berjalan di PT BRI Insurance (BRINS), khususnya pada Divisi Compliance and Risk Management (CRM). Kegiatan ini bertujuan untuk memahami alur kerja, prosedur yang berlaku, serta mengidentifikasi kebutuhan sistem yang akan dikembangkan.

2. **Wawancara**

   Wawancara dilakukan secara terstruktur dengan pihak-pihak yang terlibat dalam proses manajemen risiko, seperti staf Compliance and Risk Management (CRM) dan Project Manager. Teknik ini digunakan untuk menggali kebutuhan fungsional dan nonfungsional sistem, kendala yang dihadapi dalam metode manual, serta harapan pengguna terhadap sistem Risk Register berbasis Artificial Intelligence (AI).

3. **Studi Pustaka**

   Studi pustaka dilakukan dengan menelaah berbagai sumber literatur yang relevan, meliputi konsep manajemen risiko, pengembangan sistem informasi, framework Laravel, serta perancangan basis data MySQL. Literatur ini digunakan sebagai landasan teoritis dalam perancangan dan implementasi sistem (Otwell, 2023; Wibowo & Santosa, 2022).

4. **Dokumentasi**

   Teknik dokumentasi dilakukan dengan mengumpulkan dan menganalisis dokumen internal organisasi, termasuk file Risk Register sebelumnya yang berbasis spreadsheet.

Data tersebut digunakan sebagai referensi dalam merancang struktur data, format pencatatan risiko, serta parameter penilaian risiko pada sistem yang dikembangkan.

5. **Studi Literatur Standar dan Regulasi**

   Penelitian ini juga mengacu pada standar dan regulasi yang berlaku, antara lain ISO 31000:2018 tentang pedoman manajemen risiko serta regulasi Otoritas Jasa Keuangan (OJK) terkait penerapan manajemen risiko pada lembaga jasa keuangan (Otoritas Jasa Keuangan, 2019; 2021). Acuan tersebut digunakan untuk memastikan sistem yang dikembangkan selaras dengan prinsip tata kelola dan ketentuan regulasi yang berlaku.

3.3 Teknik Analisis Data

Teknik analisis data dalam penelitian ini menggunakan pendekatan **deskriptif kualitatif** dengan menganalisis hasil pengujian dan evaluasi sistem. Analisis dilakukan berdasarkan data yang diperoleh dari **Black Box Testing** dan **User Acceptance Testing (UAT)**.

Analisis hasil **Black Box Testing** difokuskan pada pengujian fungsionalitas sistem untuk memastikan setiap fitur berjalan sesuai dengan kebutuhan dan spesifikasi yang telah ditetapkan. Sementara itu, analisis hasil **User Acceptance Testing (UAT)** dilakukan untuk mengevaluasi tingkat penerimaan pengguna terhadap sistem yang dikembangkan.

Adapun aspek yang dianalisis meliputi:

1. **Fungsionalitas Sistem**, yaitu memastikan seluruh fitur sistem berjalan sesuai kebutuhan pengguna.
2. **Kecepatan dan Efisiensi**, yaitu menilai respons sistem dan efisiensi pemrosesan data.
3. **Keamanan Data**, yaitu mengevaluasi mekanisme autentikasi dan pengelolaan hak akses pengguna sesuai dengan prinsip manajemen risiko (Otoritas Jasa Keuangan, 2021).
4. **Kemudahan Penggunaan (*User Experience*)**, yaitu menilai tingkat kenyamanan, kemudahan navigasi, serta kemudahan penggunaan sistem oleh pengguna akhir .

Hasil analisis data ini digunakan sebagai dasar untuk menilai kelayakan sistem serta sebagai bahan evaluasi dalam pengembangan sistem di masa mendatang.

3.4 Perangkat yang Digunakan

Perangkat yang digunakan dalam penelitian ini terdiri dari perangkat keras (*hardware*) dan perangkat lunak (*software*) yang mendukung proses pengembangan, pengujian, dan evaluasi Website Risk Register berbasis web.

1. **Perangkat Keras (Hardware)**

Perangkat keras yang digunakan dalam penelitian ini meliputi:

- Laptop dengan prosesor minimal **Intel Core i5**

- **RAM 8 GB**
- **SSD 256 GB**
- Koneksi internet yang stabil

2. **Perangkat Lunak (Software)**

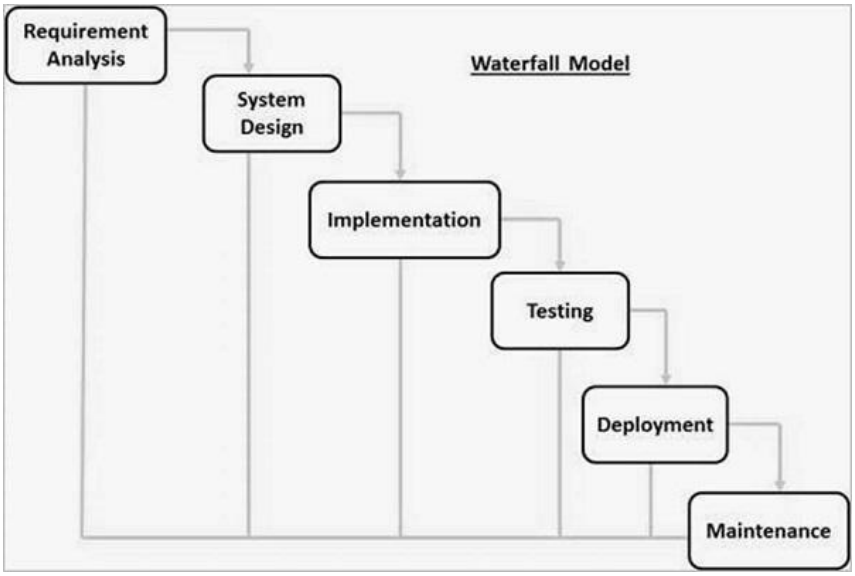
Perangkat lunak yang digunakan dalam penelitian ini meliputi:

- Sistem Operasi: **Windows 10 / Windows 11**
- Framework: **Laravel 10**
- Web Server: **XAMPP**
- Database Management System: **MySQL**
- Text Editor: **Visual Studio Code**
- Browser: **Google Chrome**

Seluruh perangkat tersebut digunakan untuk mendukung proses perancangan, implementasi, pengujian, serta evaluasi sistem agar berjalan secara optimal dan sesuai dengan kebutuhan penelitian.

3.5 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah **Waterfall Model**, yang merupakan salah satu pendekatan dalam kerangka **Software Development Life Cycle (SDLC)**. Model Waterfall dipilih karena memiliki tahapan yang sistematis, berurutan, dan terstruktur, sehingga sesuai untuk pengembangan sistem dengan kebutuhan yang telah didefinisikan secara jelas pada tahap awal (Pressman R. S., {Software Engineering: A Practitioner's Approach, 2020).



Gambar 3. 1 Waterfall Model

Tahapan pengembangan sistem dalam penelitian ini meliputi:

1. Tahap Analisis Kebutuhan

Tahap analisis kebutuhan dilakukan melalui observasi langsung dan wawancara dengan Divisi Compliance and Risk Management PT BRI Insurance (BRINS). Berdasarkan hasil analisis, diperoleh beberapa temuan utama terkait sistem pengelolaan risiko yang berjalan saat ini.

Proses pencatatan dan pengelolaan risiko masih menggunakan spreadsheet (Microsoft Excel), sehingga menimbulkan beberapa kendala, antara lain:

1. Data risiko tidak terintegrasi secara terpusat, sehingga menyulitkan monitoring dan pelaporan.
2. Penyusunan justifikasi risiko dan rekomendasi mitigasi dilakukan secara manual, yang memerlukan waktu relatif lama.
3. Tingginya ketergantungan pada subjektivitas pengguna dalam menilai risiko.
4. Potensi terjadinya duplikasi data dan kesalahan input.
5. Proses pelaporan belum otomatis dan kurang sistematis.

Berdasarkan permasalahan tersebut, dirumuskan kebutuhan sistem sebagai berikut:

a. Kebutuhan Fungsional

1. Sistem mampu mengelola data risiko secara terpusat berbasis database.
2. Sistem dapat melakukan proses Create, Read, Update, dan Delete (CRUD) data risiko.
3. Sistem dapat menghitung nilai risiko secara otomatis berdasarkan parameter Likelihood dan Impact.
4. Sistem dapat mengklasifikasikan level risiko (Low, Medium, High).
5. Sistem terintegrasi dengan Artificial Intelligence untuk menghasilkan justifikasi risiko dan rekomendasi mitigasi secara otomatis.
6. Sistem dapat menghasilkan laporan risiko secara sistematis dan terdokumentasi.
7. Sistem menerapkan Role-Based Access Control (RBAC) untuk membatasi akses berdasarkan peran pengguna.

b. Kebutuhan Non-Fungsional

1. Sistem berbasis web dan dapat diakses melalui browser.
2. Sistem memiliki keamanan data melalui autentikasi dan otorisasi pengguna.
3. Sistem memiliki performa yang stabil dan responsif.
4. Sistem mampu menyimpan data secara konsisten dan aman dalam basis data.

Tahap analisis kebutuhan ini menjadi dasar dalam perancangan sistem pada tahap berikutnya.

2. **Tahap 2 Perancangan Sistem (System Design)**

Pada tahap ini dilakukan perancangan arsitektur sistem secara menyeluruh, meliputi:

- Perancangan struktur basis data menggunakan **MySQL**
- Perancangan antarmuka pengguna (User Interface) berbasis **Laravel Blade Template**
- Pemodelan sistem menggunakan diagram **UML**, seperti *Use Case Diagram*, *Activity Diagram*, dan *Sequence Diagram*, untuk menggambarkan alur proses bisnis serta interaksi antaraktor dalam sistem (Suryawan, 2021).

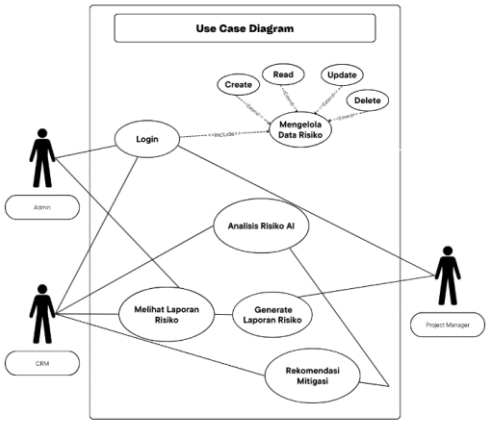
Tahap perancangan bertujuan untuk memastikan sistem yang dikembangkan sesuai dengan kebutuhan pengguna serta mendukung integrasi teknologi Artificial Intelligence (AI).

Perancangan sistem dilakukan menggunakan **Unified Modeling Language (UML)** untuk memodelkan proses bisnis, interaksi antaraktor, serta alur kerja sistem sebelum tahap implementasi. Diagram UML yang digunakan dalam penelitian ini meliputi *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Entity Relationship Diagram* (ERD).

- Use Case Diagram

Use Case Diagram menggambarkan interaksi antara pengguna dengan sistem. Aktor dalam sistem ini terdiri dari Admin, CRM, dan Project Manager, yang masing-masing memiliki hak akses dan fungsi berbeda.

Usecase diagram digunakan untuk menspesifikasikan apa yang dapat dilakukan oleh sistem atau untuk menspesifikasikan kebutuhan fungsional utama dari system (Sigi, 2021).



Gambar 3. 2 Use Case Diagram

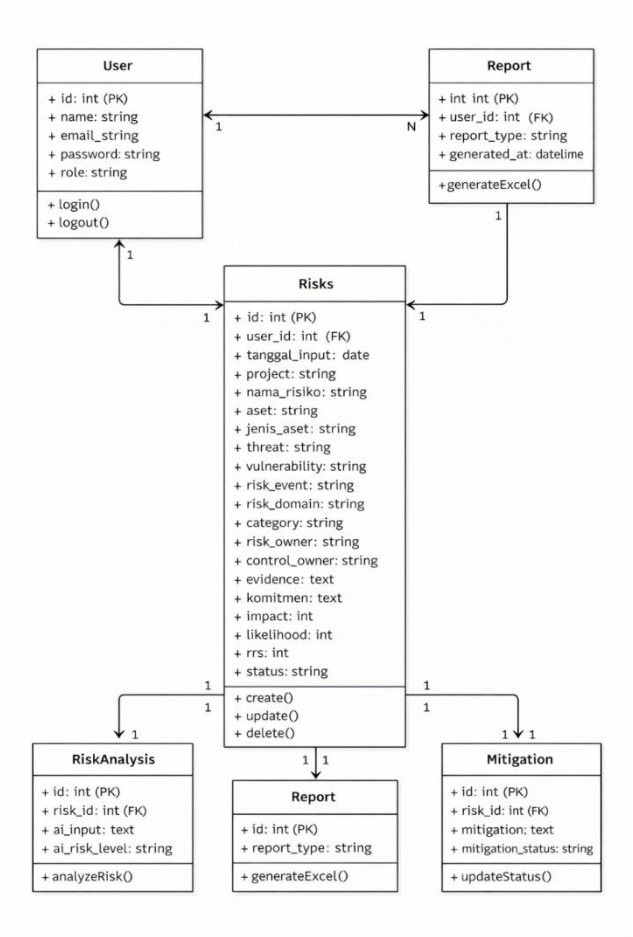
Gambar 3.1 menjelaskan *use case* diagram yang ditampilkan pada gambar 3.1 dapat dilihat bahwa admin, *CRM,PM* dapat melakukan hal yang berbeda diantaranya seperti pada tabel berikut:

Tabel 3. 1 Penjelasan Use Case Diagram

Nama Aktor	**Use Case**	**Deskripsi**	**Relasi dengan Use Case Lain**
Admin	Login	Admin melakukan autentikasi untuk masuk ke sistem	Diperlukan sebelum mengakses semua use case
Admin	Mengelola Data Risiko	Admin menambah, melihat, mengubah, dan menghapus data risiko	*Include* Create, Read, Update, Delete
Admin	Analisis Risiko AI	Admin menjalankan analisis risiko berbasis AI dari data yang ada	Menggunakan data dari Mengelola Data Risiko
Admin	Melihat Laporan Risiko	Admin melihat hasil laporan risiko	Berhubungan dengan Generate Laporan Risiko
CRM	Login	CRM melakukan autentikasi ke sistem	Diperlukan sebelum mengakses fitur
CRM	Analisis Risiko AI	CRM dapat menjalankan analisis risiko	Menggunakan data risiko yang tersedia
CRM	Melihat Laporan Risiko	CRM memantau laporan risiko	Berhubungan dengan Generate Laporan Risiko
CRM	Rekomendasi Mitigasi	CRM melihat saran mitigasi dari hasil analisis	Berasal dari Analisis Risiko AI
Project Manager	Login	PM masuk ke sistem untuk melihat informasi	Diperlukan sebelum mengakses fitur
Project Manager	Melihat Laporan Risiko	PM melihat laporan risiko sebagai bahan evaluasi	Hasil dari Generate Laporan Risiko
Project Manager	Generate Laporan Risiko	PM membuat laporan risiko	Mengambil data dari sistem
Project Manager	Rekomendasi Mitigasi	PM melihat rekomendasi penanganan risiko	Berasal dari Analisis Risiko AI

- *Class* Diagram

Selanjutnya menentukan *Class* untuk menggambarkan objek atribut, properti, dan hubungan umum antara objek dan semantik. Pada diagram ini menggambarkan seluruh class yang ada didalam sistem termasuk hubungannya.



Gambar 3. 3 Class Diagram

Gambar 3.3 menunjukkan *Class Diagram* dari Sistem Manajemen Risiko berbasis web yang dirancang. Class diagram menggambarkan struktur statis sistem yang terdiri dari kelas, atribut, metode, serta hubungan antar kelas dalam pendekatan berorientasi objek (Lestari, 2018). Diagram ini menjadi dasar dalam proses implementasi sistem karena memperlihatkan bagaimana data dan fungsi saling terintegrasi dalam satu kesatuan sistem (Pressman, 2020).

Class diagram ini terdiri dari lima kelas utama yaitu **User**, **Risks**, **Report**, **RiskAnalysis**, dan **Mitigation**.

1. Class User

Kelas *User* merepresentasikan pengguna sistem yang memiliki atribut seperti id, name, email, password, dan role. Method login() dan logout() digunakan untuk proses autentikasi pengguna. Pengelolaan hak akses berbasis peran (*role-based access control*) mendukung

keamanan sistem dan pembagian wewenang pengguna (Sandhu, 1996). Relasi antara *User* dan *Risks* bersifat one-to-many (1:N), yang berarti satu pengguna dapat mengelola lebih dari satu data risiko.

2. Class Risks

Kelas *Risks* merupakan inti sistem karena menyimpan seluruh data risiko yang diinput, seperti nama_risiko, aset, threat, vulnerability, impact, likelihood, hingga status. Perhitungan nilai risiko dilakukan berdasarkan parameter dampak (*impact*) dan kemungkinan (*likelihood*) sesuai dengan prinsip manajemen risiko (ISO, 2018). Method create(), update(), dan delete() menunjukkan proses *CRUD* dalam pengelolaan data risiko yang umum diterapkan pada sistem berbasis web (Pressman, 2020).

Kelas ini berelasi dengan:

- **User**, sebagai pihak yang menginput risiko,
- **Report**, sebagai keluaran sistem dalam bentuk laporan,
- **RiskAnalysis**, untuk analisis tingkat risiko,
- **Mitigation**, untuk tindakan pengendalian risiko.

3. Class Report

Kelas *Report* digunakan untuk menghasilkan laporan risiko dalam bentuk file Excel melalui method generateExcel(). Fitur pelaporan ini mendukung kebutuhan dalam pengambilan keputusan berbasis data (Laudon, 2020). Satu pengguna dapat menghasilkan lebih dari satu laporan (relasi 1:N).

4. Class RiskAnalysis

Kelas *RiskAnalysis* menyimpan hasil analisis risiko berbasis sistem atau kecerdasan buatan dengan atribut seperti ai_input dan ai_risk_level. Penggunaan pendekatan sistem pendukung keputusan (*Decision Support System*) membantu dalam optimasi evaluasi risiko (Hidayat, 2024). Relasinya bersifat one-to-one (1:1) dengan kelas *Risks*.

5. Class Mitigation

Kelas *Mitigation* menyimpan tindakan mitigasi risiko beserta status pelaksanaannya. Proses mitigasi merupakan bagian penting dalam siklus manajemen risiko untuk mengurangi dampak dan kemungkinan terjadinya risiko (Otoritas Jasa Keuangan, 2019). Method updateStatus() digunakan untuk memperbarui status pengendalian risiko.

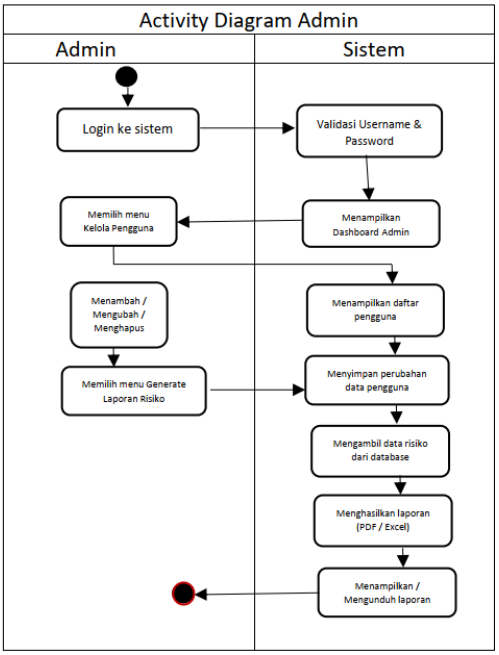
Secara keseluruhan, Class Diagram pada Gambar 3.3 menggambarkan struktur sistem manajemen risiko berbasis web yang terintegrasi, mulai dari proses input risiko, analisis, mitigasi, hingga pelaporan dalam bentuk Excel. Perancangan ini sesuai dengan prinsip

rekayasa perangkat lunak berorientasi objek dan mendukung implementasi sistem yang sistematis dan terstruktur (Pressman, 2020; Lestari, 2018).

- Activity Diagram

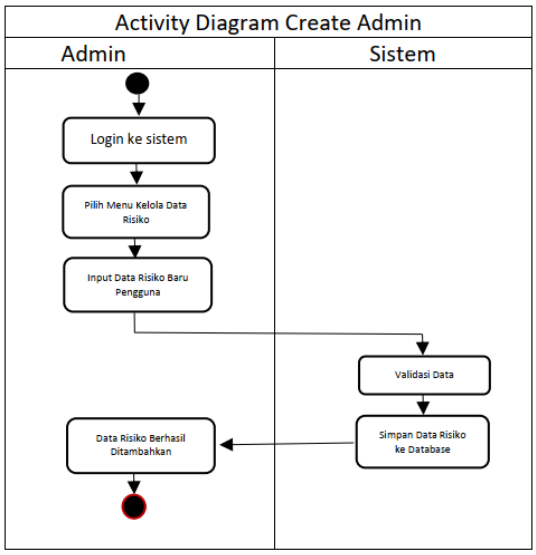
Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, keputusan (*decision*) yang mungkin terjadi, dan bagaimana alir tersebut berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi *(Haris & Nirmaya, 2023; Pressman & Maxim, 2020)*.

Pada penelitian ini, Activity Diagram digunakan untuk menggambarkan alur aktivitas pengguna dalam sistem manajemen risiko, yang meliputi proses login, pengelolaan data risiko (create, read, update, delete), analisis risiko berbasis kecerdasan buatan (AI), hingga pemberian rekomendasi mitigasi risiko secara sistematis.



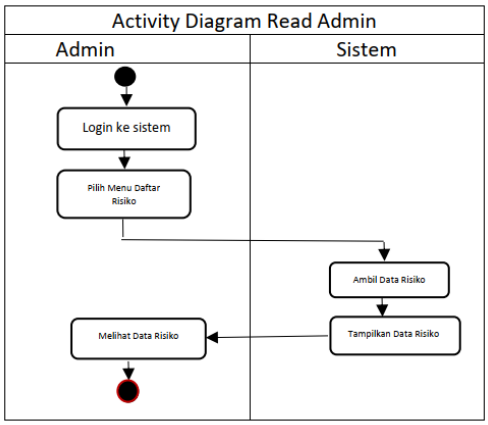
Gambar 3. 4 Activity Diagram

Activity Diagram Admin pada bagian 3.4 menjelaskan proses login, pengelolaan data kinerja, serta pembuatan laporan oleh Admin yang divalidasi dan diproses oleh sistem hingga menghasilkan output laporan dalam format Excel.



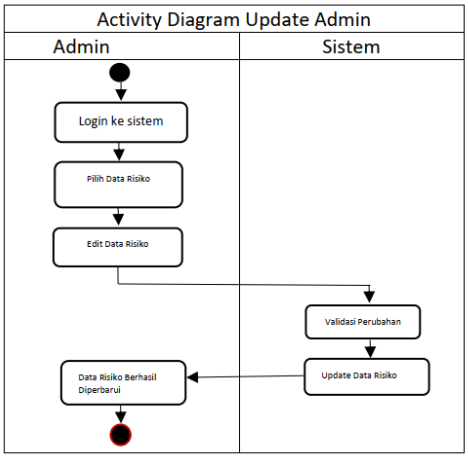
Gambar 3. 5 Activity Diagram Create Admin

Activity Diagram Create Admin pada bagian 3.5 menjelaskan alur prosesketika Admin login ke sistem, memilih menu kelola data risiko, lalu melakukan input data risiko baru yang kemudian divalidasi oleh sistem, disimpan ke database, dan ditampilkan konfirmasi bahwa data berhasil ditambahkan sebagai output akhir.



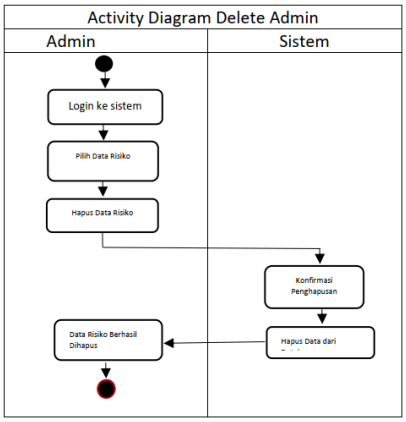
Gambar 3. 6 Activity Diagram Read Admin

Activity Diagram Read Admin pada bagian 3.6 menjelaskan alur proses ketika Admin login ke sistem, memilih menu daftar risiko, kemudian sistem mengambil data risiko dari database dan menampilkannya sehingga Admin dapat melihat data risiko sebagai output akhir.



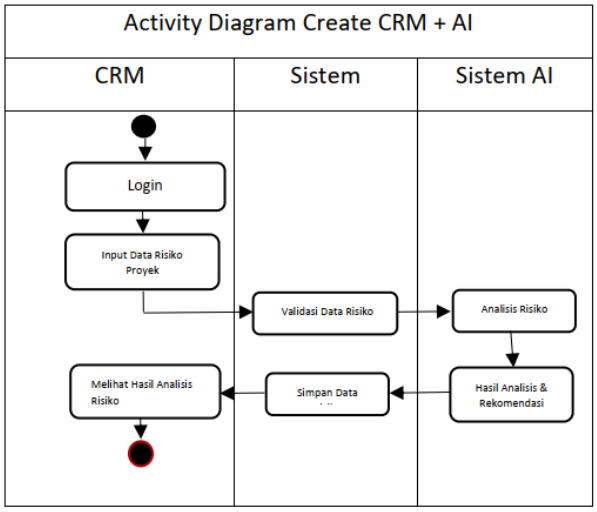
Gambar 3. 7 Activity Diagram Update Admin

Activity Diagram Update Admin pada bagian 3.7 menjelaskan alur proses ketika Admin login ke sistem, memilih data risiko, lalu melakukan edit data yang kemudian divalidasi oleh sistem, diperbarui di database, dan menghasilkan konfirmasi bahwa data risiko berhasil diperbarui sebagai output akhir.



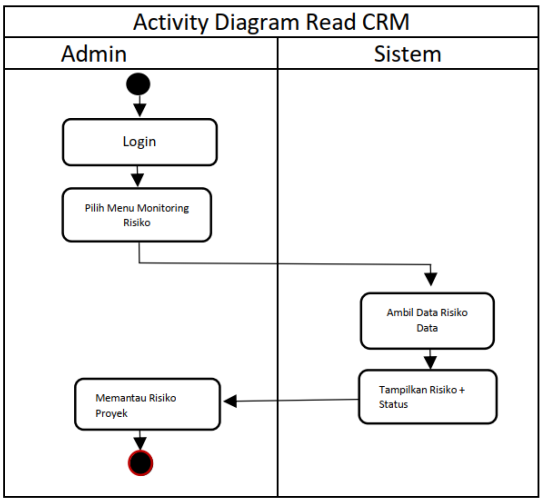
Gambar 3. 8 Activity Diagram Delete Admin

Activity Diagram Delete Admin pada bagian 3.8 menjelaskan alur proses ketika Admin login ke sistem, memilih data buku, lalu melakukan penghapusan data yang dikonfirmasi oleh sistem, kemudian sistem menghapus data dari database dan menampilkan konfirmasi bahwa data buku berhasil dihapus sebagai output akhir.



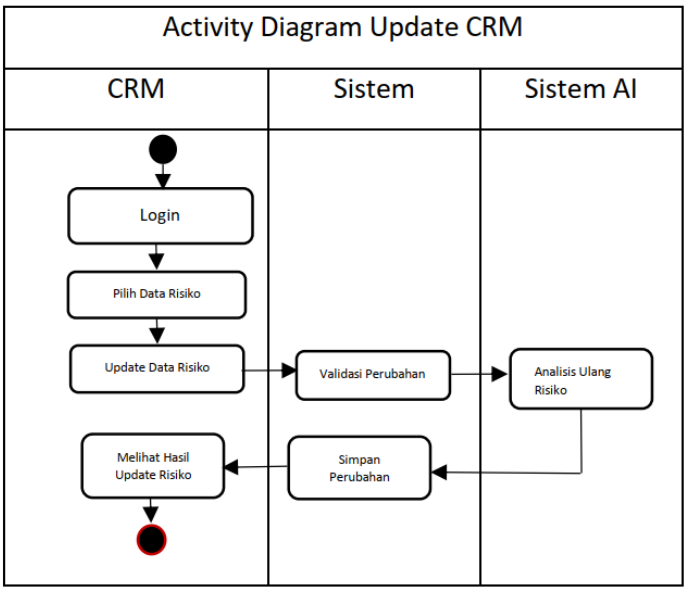
Gambar 3. 9 Activity Diagram Create CRM + AI

Activity Diagram Create CRM + AI pada bagian 3.9 menjelaskan alur proses ketika pengguna login ke sistem CRM, kemudian melakukan input data risiko proyek yang divalidasi dan disimpan oleh sistem, lalu sistem AI melakukan analisis terhadap data tersebut dan menghasilkan rekomendasi, sehingga pengguna dapat melihat hasil analisis risiko beserta saran yang diberikan sebagai output akhir.



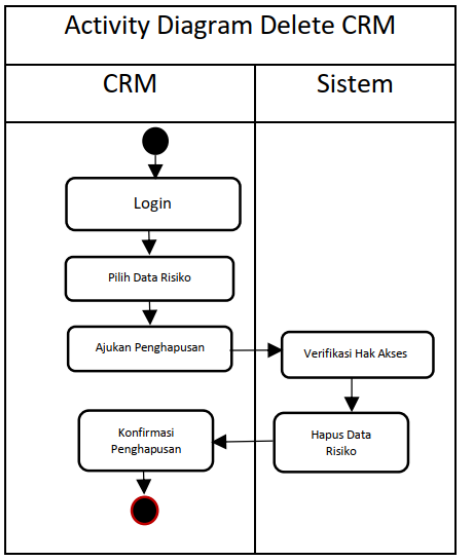
Gambar 3. 10 Activity Diagram Read CRM

Activity Diagram Read CRM pada bagian 3.10 menjelaskan alur proses ketika Admin login ke sistem, memilih menu monitoring risiko, lalu sistem mengambil data risiko dari database dan menampilkan informasi risiko beserta statusnya sehingga Admin dapat memantau risiko proyek secara langsung sebagai output akhir.



Gambar 3. 11 Activity Diagram Update CRM

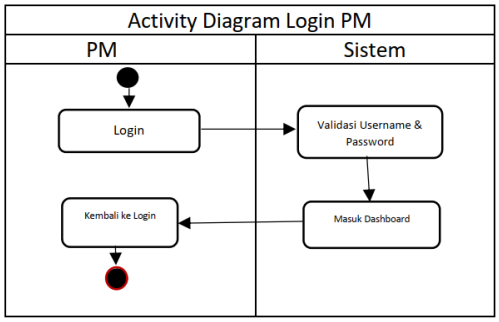
Activity Diagram Update CRM pada bagian 3.11 menjelaskan alur proses ketika pengguna login ke sistem CRM, memilih data risiko, lalu melakukan update data yang disimpan oleh sistem, kemudian sistem AI melakukan analisis ulang terhadap data tersebut sehingga pengguna dapat melihat hasil pembaruan risiko beserta rekomendasi yang dihasilkan sebagai output akhir.



Gambar 3. 12 Activity Diagram Delete CRM

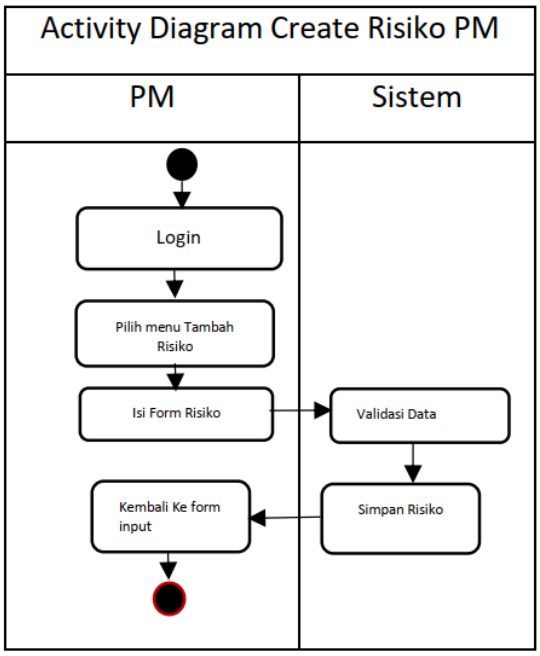
Activity Diagram Delete CRM pada bagian 3.12 menjelaskan alur proses ketika pengguna login ke sistem CRM, memilih data risiko yang ingin dihapus, kemudian mengajukan dan melakukan konfirmasi penghapusan, sementara sistem melakukan verifikasi

hak akses dan menghapus data risiko dari database, sehingga proses berakhir dengan konfirmasi bahwa data berhasil dihapus.



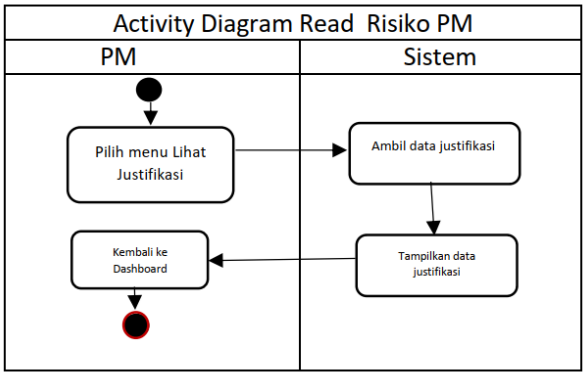
Gambar 3. 13 Activity Diagram Login PM

Activity Diagram Login PM pada bagian 3.13 menjelaskan alur proses ketika Project Manager (PM) melakukan login ke sistem, kemudian sistem memvalidasi username dan password; jika valid maka PM diarahkan masuk ke dashboard, sedangkan jika tidak valid sistem mengembalikan ke halaman login, sehingga diagram ini menggambarkan mekanisme autentikasi dan akses awal pengguna.



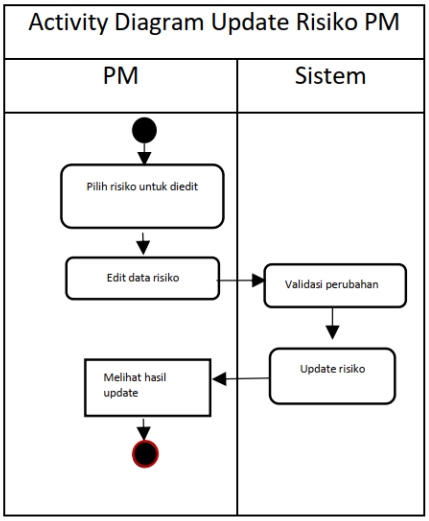
Gambar 3. 14 Activity Diagram Create Risiko PM

Activity Diagram Create Risiko PM pada bagian 3.14 menjelaskan alur proses ketika Project Manager (PM) login ke sistem, memilih menu tambah risiko, lalu mengisi form risiko yang divalidasi oleh sistem; jika data tidak valid maka dikembalikan ke form input, sedangkan jika valid sistem menyimpan data risiko sehingga proses berakhir dengan konfirmasi bahwa risiko berhasil ditambahkan.



Gambar 3. 15 Activity Diagram Read Risiko PM

Activity Diagram Read Risiko PM pada bagian 3.15 menjelaskan alur proses ketika Project Manager (PM) memilih menu lihat justifikasi, kemudian sistem mengambil dan menampilkan data justifikasi risiko, sehingga PM dapat memantau informasi tersebut sebelum kembali ke dashboard sebagai output akhir.

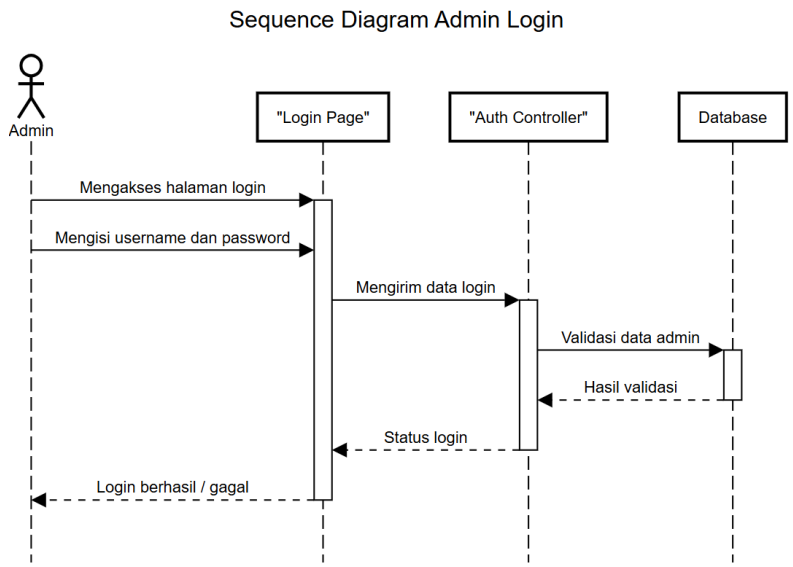


Gambar 3. 16 Activity Diagram Update Risiko PM

Activity Diagram Update Risiko PM pada bagian 3.16 menjelaskan alur proses ketika Project Manager (PM) memilih risiko yang akan diedit, melakukan perubahan data, kemudian sistem memvalidasi dan memperbarui data tersebut, sehingga PM dapat melihat hasil pembaruan risiko sebagai output akhir.

- Sequence Diagram

Sequence diagram merupakan penjelasan interaksi antar objek yang disusun dalam suatu urutan waktu, yaitu urutan kejadian yang dilakukan oleh seorang aktor dalam menjalankan sistem. Diagram ini secara khusus berasosiasi dengan *use case* dan digunakan untuk menggambarkan alur komunikasi antar objek secara detail berdasarkan waktu kejadian *(Hartanti & Desyanita, 2022; Sommerville, 2016)*.



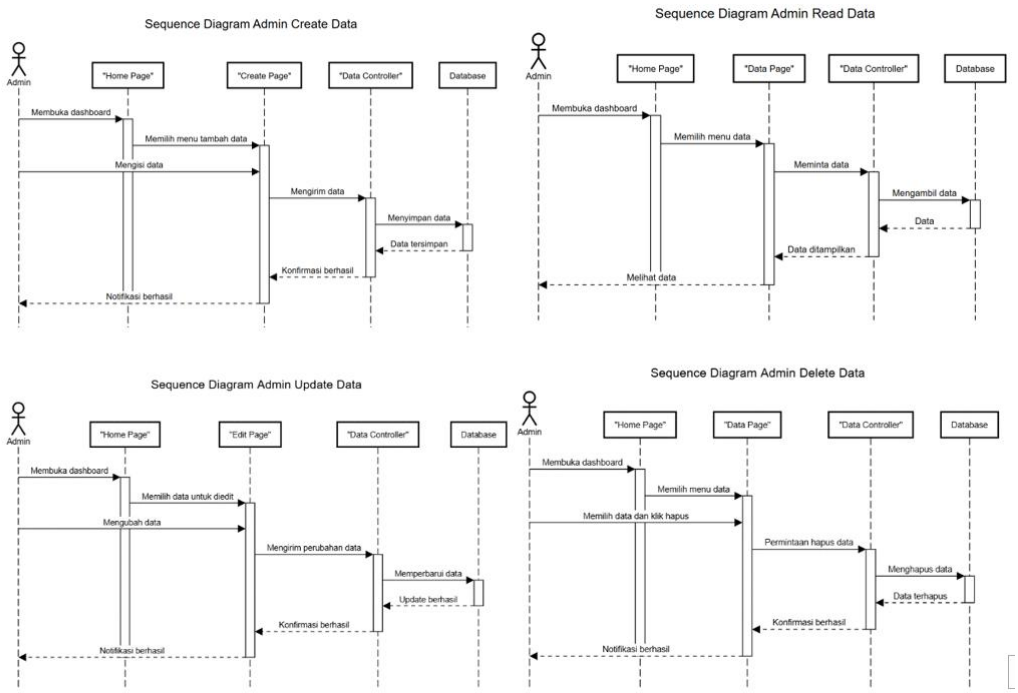
Gambar 3. 17 Sequence Diagram Admin Login

Pada Gambar 3.17 Sequence Diagram Admin Login menggambarkan alur proses autentikasi Admin ke dalam sistem. Proses diawali ketika Admin mengakses halaman login (Login Page). Selanjutnya, Admin mengisi username dan password pada form login yang tersedia.

Data login yang telah diinput kemudian dikirim oleh Login Page ke Auth Controller untuk dilakukan proses autentikasi. Auth Controller bertugas memvalidasi data login dengan cara mencocokkan username dan password yang dimasukkan dengan data Admin yang tersimpan di dalam Database.

Database akan mengembalikan hasil validasi kepada Auth Controller. Apabila data yang dimasukkan sesuai, maka sistem akan mengembalikan status login berhasil dan Admin dapat mengakses dashboard sistem. Namun, jika data login tidak sesuai, sistem akan mengembalikan status login gagal dan Admin diminta untuk mengulangi proses login.

Sequence diagram ini menunjukkan bahwa proses login Admin melibatkan beberapa komponen utama, yaitu Admin sebagai aktor, Login Page sebagai antarmuka pengguna, Auth Controller sebagai pengelola autentikasi, serta Database sebagai penyimpan data Admin.



Gambar 3. 18 Sequence Diagram Admin

Pada Gambar 3.18 Sequence Diagram Admin menjelaskan alur proses pengelolaan data oleh Admin yang meliputi proses Create, Read, Update, dan Delete (CRUD) pada sistem. Sequence diagram ini menggambarkan interaksi antara Admin sebagai aktor dengan sistem yang terdiri dari Home Page, halaman pengelolaan data, Data Controller, serta Database.

1. Sequence Diagram Admin Create Data

Proses Create Data diawali ketika Admin membuka dashboard pada Home Page dan memilih menu tambah data. Admin kemudian mengisi data pada Create Page sesuai dengan kebutuhan sistem. Data yang telah diinput akan dikirimkan ke Data Controller untuk diproses dan disimpan ke dalam Database. Setelah data berhasil disimpan, Database mengirimkan status penyimpanan ke Data Controller, kemudian sistem menampilkan notifikasi bahwa proses penambahan data berhasil dilakukan.

2. Sequence Diagram Admin Read Data

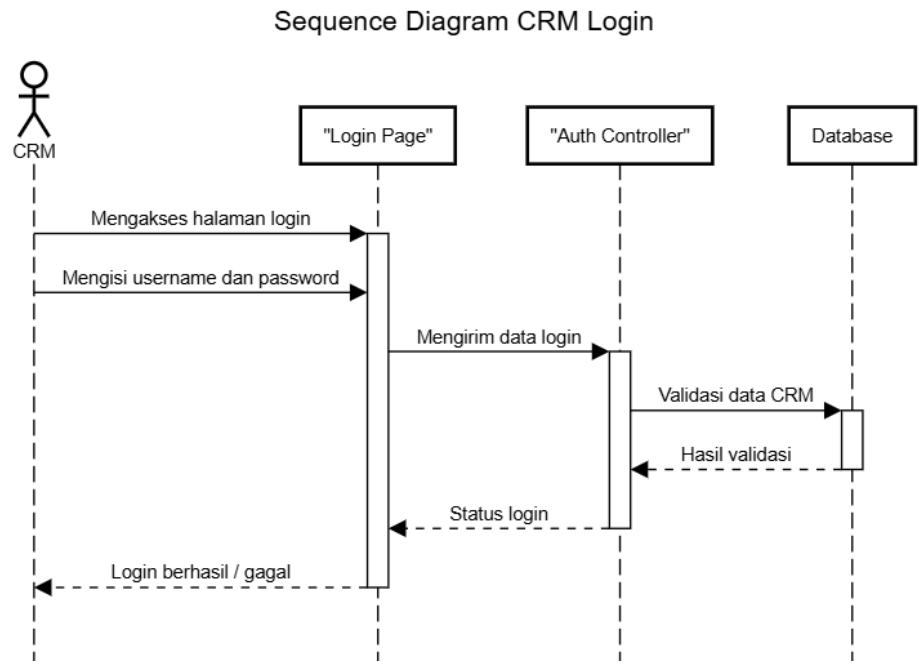
Pada proses Read Data, Admin membuka dashboard dan memilih menu data untuk melihat informasi yang tersedia. Permintaan data dikirimkan dari Data Page ke Data Controller, kemudian Data Controller mengambil data yang dibutuhkan dari Database. Setelah data berhasil diambil, sistem menampilkan data tersebut kepada Admin pada halaman data sehingga Admin dapat melihat informasi yang tersimpan di dalam sistem.

3. Sequence Diagram Admin Update Data

Proses Update Data dimulai ketika Admin membuka dashboard dan memilih data yang akan diubah. Admin melakukan perubahan data pada Edit Page, kemudian perubahan tersebut dikirimkan ke Data Controller untuk diproses. Data Controller akan memperbarui data yang ada di Database sesuai dengan perubahan yang dilakukan. Setelah proses pembaruan berhasil, sistem menampilkan notifikasi bahwa data telah berhasil diperbarui.

4. Sequence Diagram Admin Delete Data

Pada proses Delete Data, Admin membuka dashboard dan memilih data yang akan dihapus pada Data Page. Setelah Admin mengonfirmasi penghapusan data, sistem mengirimkan permintaan hapus ke Data Controller. Data Controller kemudian menghapus data tersebut dari Database. Setelah data berhasil dihapus, sistem menampilkan notifikasi kepada Admin bahwa proses penghapusan data telah berhasil dilakukan.



Gambar 3. 19 Sequence Diagram CRM Login

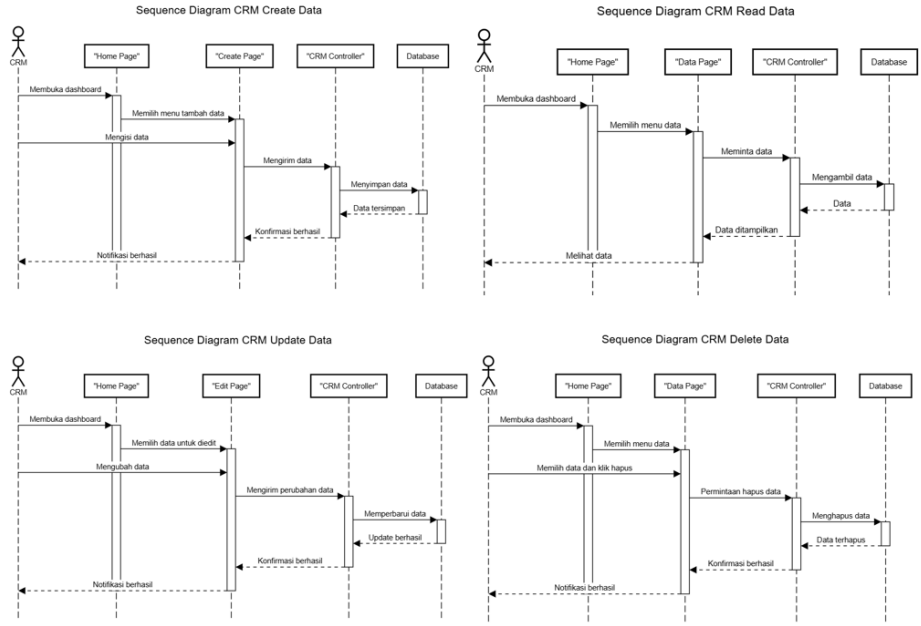
Pada Gambar 3.19 Sequence Diagram CRM Login menggambarkan alur proses autentikasi pengguna CRM ke dalam sistem. Proses dimulai ketika CRM sebagai aktor mengakses halaman login (Login Page). Selanjutnya, CRM mengisi username dan password pada form login yang tersedia.

Data login yang telah diinput kemudian dikirimkan oleh Login Page ke Auth Controller untuk dilakukan proses validasi. Auth Controller akan memeriksa kecocokan data login CRM

dengan data yang tersimpan di dalam Database. Database selanjutnya mengembalikan hasil validasi kepada Auth Controller.

Berdasarkan hasil validasi tersebut, sistem akan mengirimkan status login kepada Login Page. Apabila data login yang dimasukkan sesuai, maka CRM berhasil masuk ke dalam sistem dan dapat mengakses dashboard CRM. Sebaliknya, jika data login tidak sesuai, sistem akan menampilkan informasi bahwa proses login gagal sehingga CRM diminta untuk mengulangi proses login.

Sequence diagram ini menunjukkan bahwa proses login CRM melibatkan beberapa komponen utama, yaitu CRM sebagai aktor, Login Page sebagai antarmuka pengguna, Auth Controller sebagai pengelola autentikasi, serta Database sebagai media penyimpanan data pengguna CRM.



Gambar 3. 20 Sequence Diagram CRM

Pada Gambar 3.20 Sequence Diagram CRM menjelaskan alur proses pengelolaan data risiko oleh pengguna CRM yang meliputi proses Create, Read, Update, dan Delete (CRUD) serta proses review dan pengisian justifikasi risiko pada sistem. Sequence diagram ini menggambarkan interaksi antara CRM sebagai aktor dengan sistem yang terdiri dari Home Page, halaman pengelolaan data, CRM Controller, serta Database.

1. Sequence Diagram CRM Create Data

Proses Create Data diawali ketika CRM membuka dashboard pada Home Page dan memilih menu tambah data risiko. CRM kemudian mengisi data risiko pada Create Page sesuai dengan informasi yang dibutuhkan. Data yang telah diinput dikirimkan ke CRM Controller

untuk diproses dan disimpan ke dalam Database. Setelah data berhasil disimpan, sistem menampilkan notifikasi bahwa proses penambahan data risiko berhasil dilakukan.

2. Sequence Diagram CRM Read Data

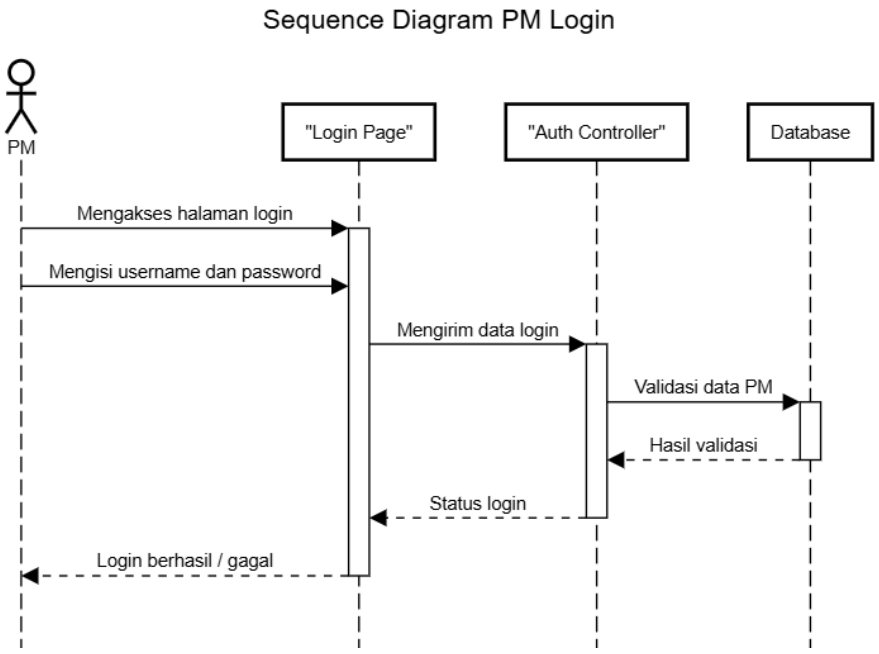
Pada proses Read Data, CRM membuka dashboard dan memilih menu data untuk melihat daftar risiko yang tersedia. Permintaan data dikirimkan dari Data Page ke CRM Controller, kemudian CRM Controller mengambil data risiko dari Database. Setelah data berhasil diambil, sistem menampilkan data tersebut sehingga dapat ditinjau oleh CRM.

3. Sequence Diagram CRM Update Data (Review dan Justifikasi Risiko)

Proses Update Data dimulai ketika CRM memilih data risiko yang akan direview. CRM melakukan proses peninjauan risiko serta mengisi atau memperbarui justifikasi risiko pada Edit Page. Perubahan dan justifikasi yang telah diinput kemudian dikirimkan ke CRM Controller untuk diproses. CRM Controller akan memperbarui data risiko dan justifikasi yang tersimpan di dalam Database. Setelah proses pembaruan berhasil, sistem menampilkan notifikasi bahwa data dan justifikasi risiko telah berhasil diperbarui.

4. Sequence Diagram CRM Delete Data

Pada proses Delete Data, CRM membuka dashboard dan memilih data risiko yang akan dihapus pada Data Page. Setelah CRM melakukan konfirmasi penghapusan, sistem mengirimkan permintaan hapus data ke CRM Controller. CRM Controller kemudian menghapus data risiko tersebut dari Database. Setelah data berhasil dihapus, sistem menampilkan notifikasi kepada CRM bahwa proses penghapusan data telah berhasil dilakukan.



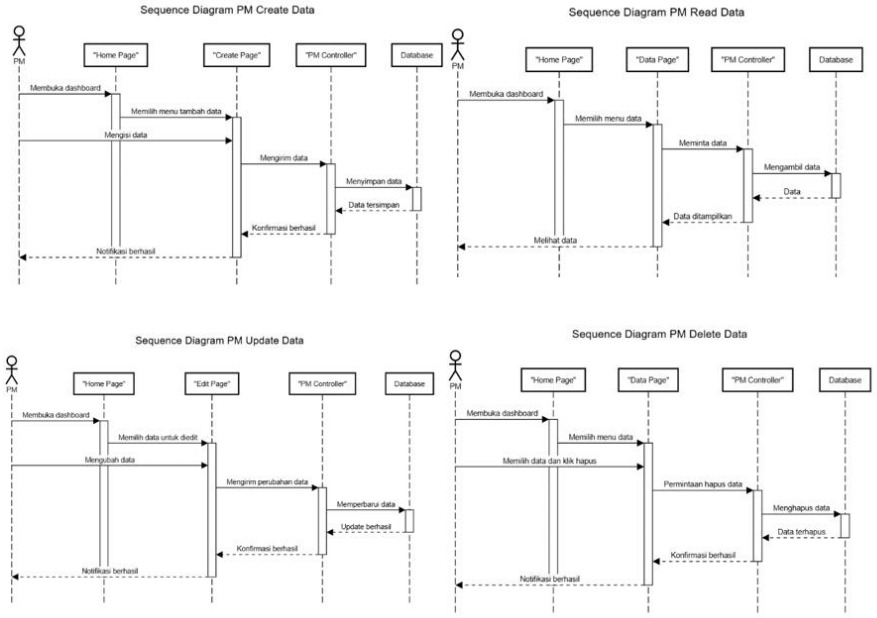
Gambar 3. 21 Sequence Diagram PM Login

Pada Gambar 3.21 Sequence Diagram PM Login menggambarkan alur proses autentikasi Project Manager (PM) ke dalam sistem. Proses dimulai ketika PM mengakses halaman login (Login Page). Selanjutnya, PM mengisi username dan password pada form login yang tersedia.

Data login yang telah diinput kemudian dikirimkan oleh Login Page ke Auth Controller untuk dilakukan proses validasi. Auth Controller akan melakukan pemeriksaan kecocokan data login PM dengan data yang tersimpan di dalam Database. Database selanjutnya mengembalikan hasil validasi kepada Auth Controller.

Berdasarkan hasil validasi tersebut, sistem akan mengirimkan status login kepada Login Page. Apabila data login yang dimasukkan sesuai, maka PM berhasil masuk ke dalam sistem dan dapat mengakses dashboard PM. Namun, jika data login tidak sesuai, sistem akan menampilkan informasi bahwa proses login gagal sehingga PM diminta untuk mengulangi proses login.

Sequence diagram ini menunjukkan bahwa proses login PM melibatkan beberapa komponen utama, yaitu PM sebagai aktor, Login Page sebagai antarmuka pengguna, Auth Controller sebagai pengelola autentikasi, serta Database sebagai media penyimpanan data PM.



Gambar 3. 22 Sequence Diagram PM

Pada Gambar 3.22 Sequence Diagram PM menjelaskan alur proses pengelolaan data risiko oleh pengguna **Project Manager (PM)** yang meliputi proses **Create, Read, Update, dan Delete (CRUD)** data risiko pada sistem. Berbeda dengan CRM, PM **tidak memiliki akses untuk melakukan review maupun pengisian justifikasi risiko**. Sequence diagram ini menggambarkan interaksi antara PM sebagai aktor dengan sistem yang terdiri dari **Home Page**, halaman pengelolaan data, **PM Controller**, serta **Database**.

1. Sequence Diagram PM Create Data

Proses Create Data diawali ketika PM membuka dashboard pada **Home Page** dan memilih menu tambah data risiko. Selanjutnya, PM mengisi data risiko pada **Create Page** sesuai dengan informasi yang dibutuhkan. Data yang telah diinput kemudian dikirimkan ke **PM Controller** untuk diproses dan disimpan ke dalam **Database**. Setelah data berhasil disimpan, sistem menampilkan notifikasi bahwa proses penambahan data risiko telah berhasil dilakukan.

2. Sequence Diagram PM Read Data

Pada proses Read Data, PM membuka dashboard dan memilih menu data untuk melihat daftar risiko yang tersedia. Permintaan data dikirimkan dari **Data Page** ke **PM Controller**, kemudian PM Controller mengambil data risiko dari **Database**. Setelah data berhasil diambil, sistem menampilkan data risiko sehingga dapat dilihat dan ditinjau oleh PM.

3. Sequence Diagram PM Update Data

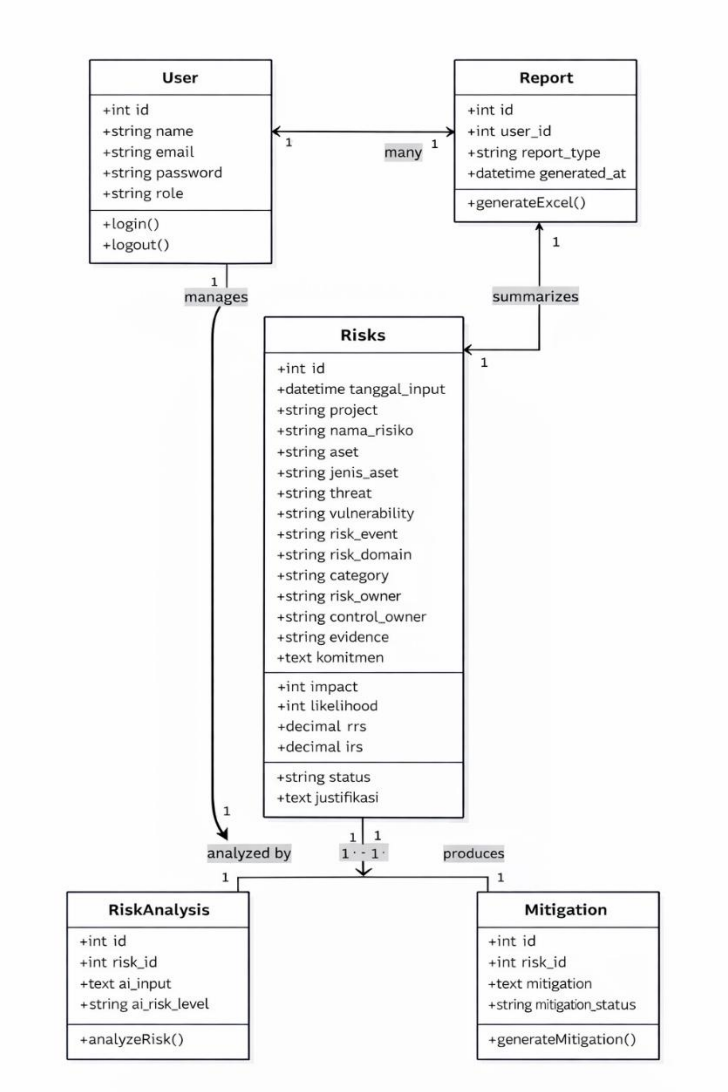
Proses Update Data dimulai ketika PM memilih data risiko yang akan diperbarui pada **Data Page**. Sistem kemudian menampilkan **Edit Page** yang berisi data risiko yang dipilih. PM melakukan perubahan terhadap data risiko sesuai kebutuhan, tanpa melakukan pengisian justifikasi risiko. Perubahan data tersebut dikirimkan ke **PM Controller** untuk diproses dan diperbarui di dalam **Database**. Setelah proses pembaruan berhasil, sistem menampilkan notifikasi bahwa data risiko telah berhasil diperbarui.

4. Sequence Diagram PM Delete Data

Pada proses Delete Data, PM membuka dashboard dan memilih data risiko yang akan dihapus pada **Data Page**. Setelah PM melakukan konfirmasi penghapusan, sistem mengirimkan permintaan penghapusan data ke **PM Controller**. PM Controller kemudian menghapus data risiko dari **Database**. Setelah data berhasil dihapus, sistem menampilkan notifikasi kepada PM bahwa proses penghapusan data risiko telah berhasil dilakukan.

- Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) digunakan untuk menggambarkan struktur basis data serta hubungan antarentitas dalam sistem **Website Risk Register berbasis Artificial Intelligence (AI)**. ERD menunjukkan entitas utama beserta atribut, *primary key*, dan *foreign key* yang saling terhubung untuk mendukung pengelolaan data risiko secara terstruktur. Dengan adanya ERD, perancangan basis data menjadi lebih sistematis, meminimalkan redundansi data, serta mendukung proses **CRUD (Create, Read, Update, Delete)** dan integrasi AI dalam pengelolaan risiko.



Gambar 3. 23 Entity Relationship Diagram

Gambar 3.23 menunjukkan *Entity Relationship Diagram* (ERD) yang menggambarkan struktur basis data pada Sistem *Intelligence Risk Register*. ERD digunakan sebagai model konseptual untuk mendeskripsikan entitas, atribut, serta hubungan antar entitas sebelum sistem diimplementasikan ke dalam basis data relasional berbasis web. Pemodelan ini berfungsi untuk memastikan bahwa struktur data yang dirancang telah sesuai dengan kebutuhan sistem serta mampu menjaga konsistensi dan integritas data.

Menurut Lestari (2018), pemodelan sistem menggunakan pendekatan terstruktur seperti UML dan ERD bertujuan memberikan gambaran konseptual yang sistematis mengenai hubungan antar data dalam suatu sistem informasi. Selain itu, Pressman (2020) menegaskan bahwa perancangan basis data yang baik merupakan bagian penting dalam rekayasa perangkat lunak karena berpengaruh terhadap efisiensi, konsistensi, dan keberlanjutan sistem.

Dalam ERD Sistem *Intelligence Risk Register* terdapat lima entitas utama, yaitu **User**, **Risks**, **Report**, **RiskAnalysis**, dan **Mitigation**, yang saling terhubung melalui relasi *One-to-Many* (1:N) dan *One-to-One* (1:1).

1. Entitas User

Entitas **User** menyimpan data pengguna yang memiliki hak akses terhadap sistem. Atribut yang dimiliki meliputi:

- id (Primary Key)
- name
- email
- password
- role

Primary Key berfungsi sebagai identitas unik bagi setiap pengguna. Atribut *role* diterapkan untuk mendukung konsep *Role-Based Access Control* (RBAC), yaitu mekanisme pengelolaan hak akses berdasarkan peran pengguna dalam sistem. Model ini memungkinkan pembatasan akses sesuai tanggung jawab masing-masing pengguna (Sandhu et al., 1996).

Relasi yang terbentuk:

- **User → Risks (1:N)**: satu pengguna dapat mengelola banyak data risiko.
- **User → Report (1:N)**: satu pengguna dapat menghasilkan beberapa laporan.

Relasi ini menunjukkan bahwa pengguna bertindak sebagai penginput dan pengelola data risiko serta pembuat laporan.

2. Entitas Risks

Entitas **Risks** merupakan entitas inti dalam sistem karena memuat seluruh data risiko yang teridentifikasi. Atribut yang terdapat pada entitas ini meliputi:

- id (Primary Key)
- tanggal_input
- project
- nama_risiko
- aset
- jenis_aset
- threat
- vulnerability
- risk_event
- risk_domain
- category

- risk_owner
- control_owner
- evidence
- komitmen
- impact
- likelihood
- rrs
- irs
- status
- justifikasi

Atribut *impact* dan *likelihood* digunakan dalam proses penilaian tingkat risiko. Penilaian ini sesuai dengan prinsip manajemen risiko dalam standar ISO 31000 yang menyatakan bahwa tingkat risiko ditentukan berdasarkan kemungkinan terjadinya suatu peristiwa dan besarnya dampak yang ditimbulkan (International Organization for Standardization, 2018).

Penerapan sistem *risk register* berbasis digital juga terbukti meningkatkan efektivitas monitoring dan dokumentasi risiko secara terstruktur dan terdokumentasi dengan baik (Rahmawati, 2023).

3. Entitas Report

Entitas **Report** menyimpan data laporan yang dihasilkan sistem berdasarkan kumpulan risiko tertentu. Atribut yang dimiliki antara lain:

- id (Primary Key)
- user_id (Foreign Key)
- report_type
- generated_at

Relasi yang terbentuk:

- **Report → Risks (1:N)**: satu laporan dapat merangkum beberapa risiko.
- **Report → User (N:1)**: setiap laporan dibuat oleh satu pengguna.

Fitur pelaporan ini mendukung konsep *Decision Support System* (DSS), di mana sistem membantu proses pengambilan keputusan melalui penyajian informasi yang terstruktur dan relevan (Hidayat, 2024).

4. Entitas RiskAnalysis

Entitas **RiskAnalysis** menyimpan hasil analisis risiko yang dilakukan menggunakan pendekatan *Artificial Intelligence* (AI).

Relasi:

- **Risks → RiskAnalysis (1:1)**: setiap risiko memiliki satu hasil analisis AI.

Pendekatan ini memungkinkan sistem memberikan evaluasi risiko secara lebih adaptif dan efisien dengan memanfaatkan teknologi kecerdasan buatan modern (Google DeepMind, 2023). Integrasi AI dalam manajemen risiko meningkatkan akurasi analisis serta mendukung proses pengambilan keputusan berbasis data.

5. Entitas Mitigation

Entitas **Mitigation** menyimpan informasi mengenai tindakan pengendalian atau respons terhadap risiko yang telah diidentifikasi.

Relasi:

- **Risks → Mitigation (1:1)**: setiap risiko memiliki satu tindakan mitigasi utama.

Konsep mitigasi ini sejalan dengan PMBOK Guide yang menyatakan bahwa setiap risiko yang teridentifikasi harus memiliki strategi respons yang jelas, baik berupa penghindaran, pengurangan, transfer, maupun penerimaan risiko (Project Management Institute, 2021).

6. Analisis Perancangan Basis Data

Perancangan ERD pada Sistem *Intelligence Risk Register* bertujuan untuk:

1. Meminimalkan redundansi data.
2. Menjaga integritas referensial melalui penggunaan Primary Key dan Foreign Key.
3. Mendukung skalabilitas dan pengembangan sistem di masa mendatang.

Menurut Laudon dan Laudon (2020), sistem informasi yang dirancang dengan struktur data yang baik akan meningkatkan kualitas pengolahan informasi dan mendukung pengambilan keputusan organisasi secara efektif.

Dengan demikian, desain ERD yang ditampilkan pada Gambar 3.23 telah memenuhi prinsip perancangan basis data relasional modern dan mendukung implementasi manajemen risiko berbasis web secara terintegrasi, sistematis, dan berorientasi pada pengambilan keputusan.

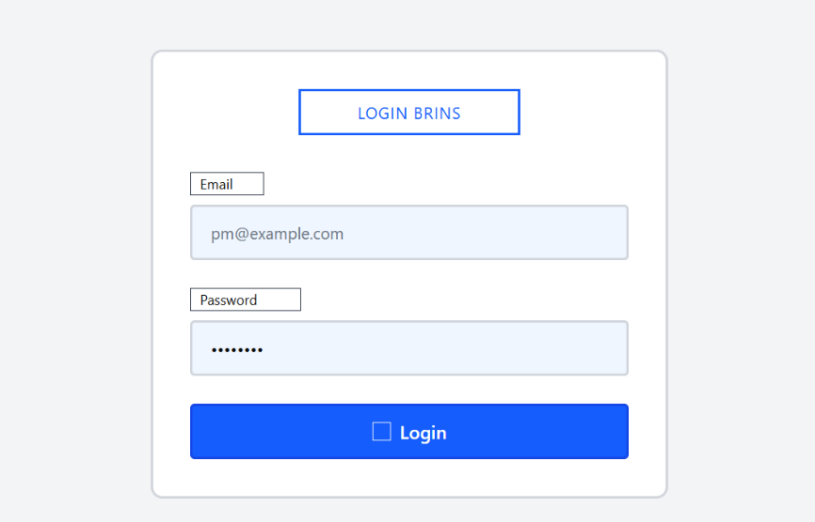
- **Desain Sistem**

Desain sistem pada penelitian ini berfokus pada **perancangan awal sistem** sebagai acuan sebelum tahap implementasi dilakukan. Desain sistem disusun berdasarkan hasil analisis kebutuhan dan digunakan sebagai pedoman dalam pengembangan sistem pada tahap implementasi.

Pada tahap ini, rancangan sistem dituangkan dalam bentuk model sistem serta **rancangan antarmuka pengguna (wireframe)**. Wireframe digunakan untuk menggambarkan struktur tampilan dan alur penggunaan sistem tanpa menampilkan detail implementasi teknis.

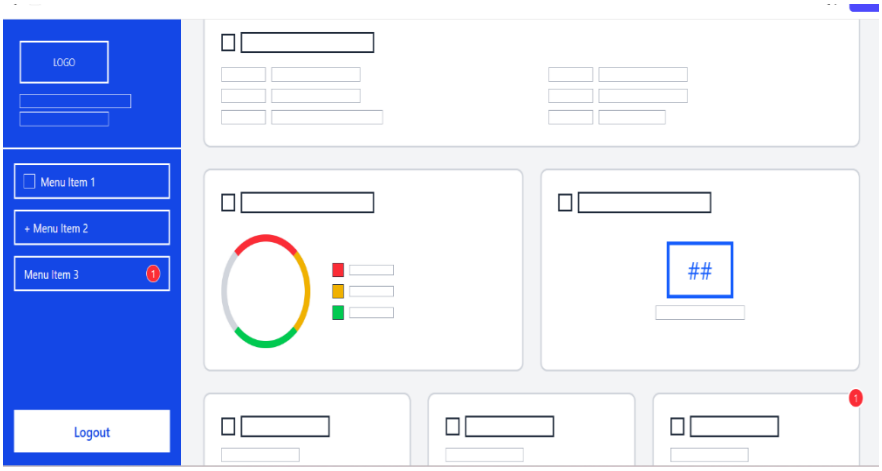
Adapun wireframe yang dirancang dalam penelitian ini meliputi:

1. **Wireframe Halaman Login**, sebagai rancangan akses awal pengguna ke dalam sistem sesuai hak akses.



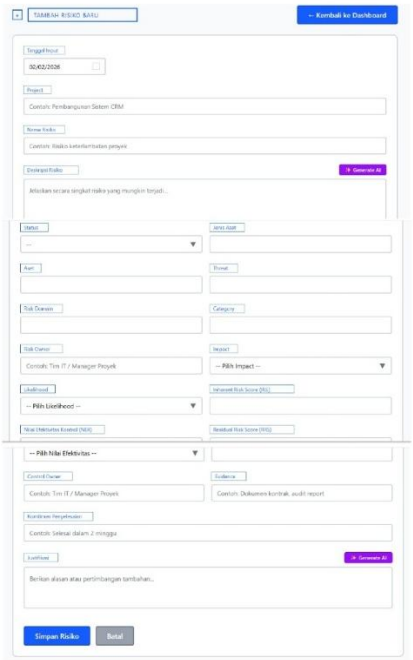
Gambar 3. 24 Wireframe Halaman Login

2. **Wireframe Halaman Dashboard**, sebagai rancangan tampilan utama yang menyajikan ringkasan informasi risiko.



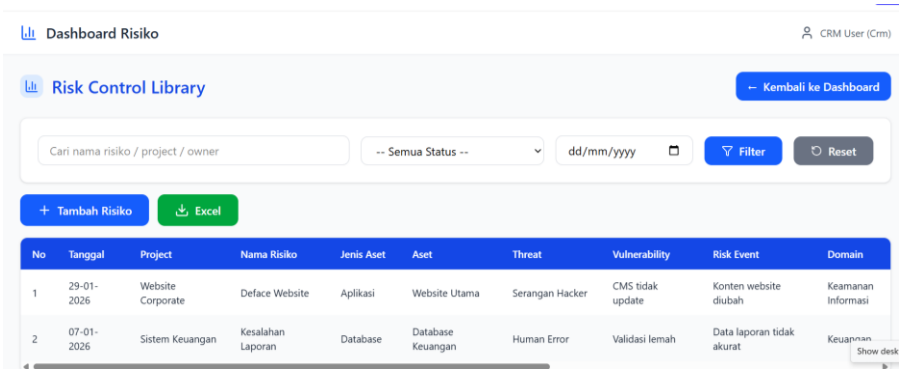
Gambar 3. 25 Wireframe Halaman Dashboard

3. **Wireframe Halaman Tambah Risiko**, sebagai rancangan proses input data risiko.



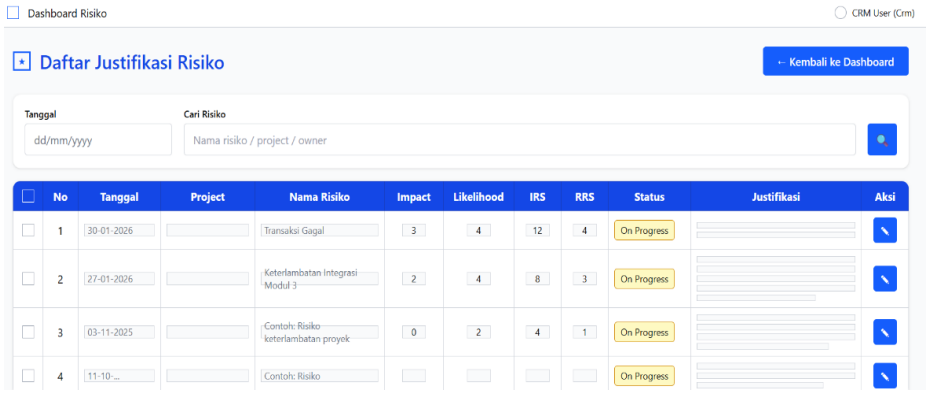
Gambar 3. 26 Wireframe Halaman Tambah Risiko

4. **Wireframe Halaman Risk Control Library (RCL)**, sebagai rancangan pengelolaan dan peninjauan kontrol risiko.



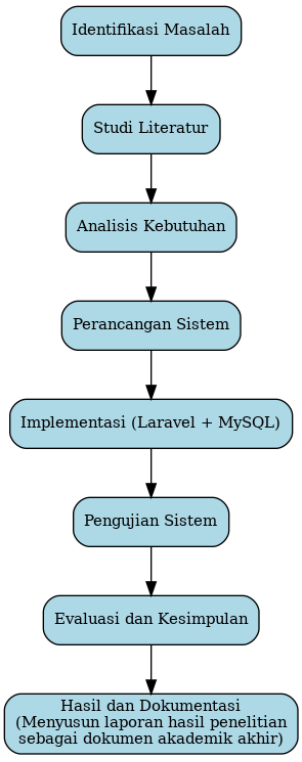
Gambar 3. 27 Wireframe Halaman Risk Control Library (RCL)

5. **Wireframe Halaman Pengisian Justifikasi Risiko**, sebagai rancangan proses review dan pengisian justifikasi oleh pengguna yang berwenang.



Gambar 3. 28 Wireframe Halaman Pengisian Justifikasi Risiko

Seluruh rancangan wireframe ini digunakan sebagai **acuan implementasi sistem** yang akan dibahas secara rinci pada **Bab IV**, sehingga tidak menampilkan hasil implementasi, melainkan gambaran konseptual dari sistem yang akan dikembangkan.



Gambar 3. 29 Prosedur Penelitian

Gambar 3.29 memperlihatkan prosedur penelitian yang menggambarkan tahapan kerja pengembangan sistem secara sistematis dan terstruktur, sehingga penelitian dapat direplikasi serta menghasilkan temuan yang dapat dipertanggungjawabkan. Prosedur penelitian ini mengacu pada metode Software Development Life Cycle (SDLC) model Waterfall

sebagaimana dijelaskan oleh Pressman (2020), yang kemudian disesuaikan dengan konteks pengembangan Website Risk Register berbasis Artificial Intelligence (AI). Model Waterfall dipilih karena memiliki tahapan yang jelas dan terstruktur, sehingga sesuai untuk pengembangan sistem yang kebutuhan awalnya telah terdefinisi dengan baik.

Adapun tahapan dalam prosedur penelitian ini meliputi:

1. **Identifikasi Masalah**

   Tahapan awal dilakukan untuk mengidentifikasi permasalahan dalam proses pencatatan dan pengelolaan risiko proyek yang masih dilakukan secara manual di lingkungan PT BRI Insurance (BRINS). Identifikasi ini bertujuan untuk menentukan kebutuhan sistem yang mampu meningkatkan efektivitas dan efisiensi manajemen risiko.

2. **Studi Literatur**

   Pengumpulan referensi dari jurnal ilmiah, buku, serta dokumentasi teknis yang relevan dengan pengembangan sistem informasi berbasis web, Laravel Framework, konsep Large Language Model (LLM), serta standar manajemen risiko ISO 31000:2018 dan pedoman Project Management Institute (PMI, 2021).

3. **Analisis Kebutuhan**

   Analisis kebutuhan dilakukan untuk menentukan kebutuhan fungsional dan non-fungsional sistem berdasarkan hasil observasi, wawancara, serta studi proses bisnis. Tahapan ini juga mencakup analisis kebutuhan integrasi API Google Gemini sebagai modul analisis risiko berbasis AI.

4. **Perancangan Sistem**

   Tahap perancangan meliputi penyusunan arsitektur sistem, struktur basis data, dan antarmuka pengguna. Pada tahap ini digunakan pemodelan UML seperti Use Case Diagram, Activity Diagram, Sequence Diagram, serta Entity Relationship Diagram (ERD) untuk menggambarkan alur sistem dan interaksi antaraktor.
   Sistem dirancang dengan pembagian peran pengguna, yaitu:

   - **Admin**: Mengelola data pengguna dan pengaturan sistem
   - **CRM**: Mencatat dan memantau risiko proyek digital
   - **Project Manager**: Memvalidasi, mengevaluasi, dan menindaklanjuti risiko

5. **Implementasi Sistem**

   Implementasi dilakukan dengan mengembangkan Website Risk Register menggunakan Laravel Framework dan MySQL Database Management System pada server lokal

XAMPP. Tahapan ini juga mencakup integrasi Google Gemini AI API untuk mendukung fitur analisis dan rekomendasi risiko secara otomatis.

6. **Pengujian Sistem**

   Pengujian sistem dilakukan menggunakan metode Black Box Testing untuk memastikan setiap fitur berjalan sesuai dengan spesifikasi yang telah ditetapkan serta bebas dari kesalahan logika dan fungsional.

7. **Evaluasi dan Kesimpulan**

   Evaluasi dilakukan dengan membandingkan hasil implementasi sistem terhadap kebutuhan awal yang telah ditentukan. Tahapan ini bertujuan untuk menilai peningkatan efisiensi, akurasi, dan efektivitas manajemen risiko dibandingkan metode manual.

8. **Hasil dan Dokumentasi**

   Tahap akhir berupa penyusunan laporan hasil penelitian dan dokumentasi sistem sebagai bentuk pertanggungjawaban akademik dan dasar pengembangan lanjutan di masa mendatang.

3.8 Metode Evaluasi Sistem

Metode evaluasi sistem yang digunakan dalam penelitian ini adalah **User Acceptance Testing (UAT)**. UAT bertujuan untuk mengetahui tingkat penerimaan pengguna terhadap Website Risk Register berbasis Artificial Intelligence (AI) yang telah dikembangkan, serta menilai kesesuaian sistem dengan kebutuhan pengguna di lingkungan PT BRI Insurance (BRINS).

Evaluasi dilakukan dengan melibatkan pengguna akhir sistem yang terdiri dari **Admin, CRM, dan Project Manager**, sesuai dengan peran dan hak akses masing-masing dalam sistem. Pengguna diminta untuk mencoba sistem berdasarkan skenario penggunaan yang telah ditentukan, kemudian memberikan penilaian melalui instrumen evaluasi berupa kuesioner.

Aspek yang dievaluasi dalam User Acceptance Testing meliputi:

1. **Kesesuaian fitur sistem** dengan kebutuhan pengguna.
2. **Kemudahan penggunaan sistem (usability)**.
3. **Kejelasan dan kenyamanan tampilan antarmuka pengguna**.
4. **Manfaat sistem** dalam mendukung proses pencatatan, pemantauan, dan pengelolaan risiko.

Hasil User Acceptance Testing digunakan sebagai dasar untuk menentukan tingkat kelayakan sistem serta sebagai bahan evaluasi untuk pengembangan dan penyempurnaan sistem di masa mendatang.

3. **Tahap 3 Implementasi**

Tahap implementasi merupakan tahap ketiga dalam model pengembangan perangkat lunak Waterfall setelah tahap analisis kebutuhan dan perancangan sistem. Pada tahap ini, seluruh rancangan sistem direalisasikan ke dalam bentuk aplikasi berbasis web yang dapat dijalankan secara fungsional sesuai dengan kebutuhan pengguna.

Implementasi sistem dilakukan menggunakan framework Laravel dengan arsitektur Model-View-Controller (MVC) dan database MySQL sebagai media penyimpanan data. Selain itu, sistem diintegrasikan dengan teknologi Artificial Intelligence berbasis Large Language Model (LLM) melalui mekanisme Application Programming Interface (API) untuk mengotomatisasi proses penyusunan justifikasi risiko dan rekomendasi mitigasi.

Proses implementasi dilakukan secara sistematis sebagai berikut:

1. **Implementasi Struktur Database**

Tahap awal implementasi dimulai dengan pembangunan database sesuai dengan Entity Relationship Diagram (ERD) yang telah dirancang. Struktur database dirancang untuk mendukung kebutuhan pencatatan risiko secara terpusat, konsisten, dan terintegrasi.
Beberapa tabel utama yang diimplementasikan meliputi:

- users
- roles
- risks
- risk_categories
- likelihoods
- impacts
- mitigations

Setiap tabel dibuat menggunakan migration pada Laravel untuk memastikan struktur dapat terdokumentasi dan direplikasi dengan baik. Relasi antar tabel diimplementasikan menggunakan foreign key guna menjaga integritas data dan mencegah inkonsistensi.
Sebagai contoh:

- Tabel risks memiliki relasi dengan users (sebagai penginput risiko).
- Tabel risks terhubung dengan likelihoods dan impacts untuk perhitungan nilai risiko.
- Relasi one-to-many diterapkan agar satu user dapat memiliki banyak data risiko.

Dengan implementasi ini, sistem tidak lagi bergantung pada file Excel terpisah, melainkan menggunakan database terpusat yang lebih aman dan terstruktur.

2. **Pengembangan Backend Menggunakan Laravel**

Backend dikembangkan menggunakan framework Laravel karena mendukung arsitektur MVC (Model-View-Controller) yang terorganisir dan mudah dipelihara.
Pada tahap ini dilakukan:

- Pembuatan **Model** untuk setiap entitas database.
- Pembuatan **Controller** untuk menangani logika bisnis.
- Konfigurasi **Routing** untuk mengatur alur request.
- Implementasi **Middleware** untuk pembatasan akses.

Controller menjadi pusat logika sistem, terutama dalam:

- Proses CRUD risiko
- Perhitungan nilai risiko otomatis
- Klasifikasi level risiko
- Integrasi API AI

Setiap input yang diterima sistem divalidasi menggunakan fitur validation Laravel untuk mencegah kesalahan data (misalnya nilai likelihood dan impact harus berupa angka dalam rentang tertentu).
Pendekatan ini memastikan backend bersifat robust, aman, dan maintainable.

3. **Pengembangan Frontend (Antarmuka Pengguna)**

Frontend dikembangkan menggunakan Blade Template Laravel agar terintegrasi langsung dengan backend. Antarmuka dirancang dengan prinsip:

- User-friendly
- Responsif
- Minimalis dan informatif

Halaman yang diimplementasikan meliputi:

- Dashboard monitoring risiko
- Form input risiko
- Halaman detail risiko
- Halaman edit dan update
- Halaman hasil justifikasi AI

Dashboard menampilkan ringkasan jumlah risiko berdasarkan level (Low, Medium, High) sehingga manajemen dapat dengan cepat memahami kondisi risiko perusahaan.
Dengan implementasi frontend ini, proses input dan monitoring risiko menjadi lebih sistematis dibandingkan metode manual sebelumnya.

4. **Implementasi Perhitungan Otomatis Nilai Risiko**

Salah satu fitur utama yang diimplementasikan adalah otomatisasi perhitungan nilai risiko. Sistem secara otomatis menghitung nilai risiko menggunakan rumus:

Risk Score = Likelihood × Impact

Perhitungan dilakukan di dalam controller sehingga ketika user menyimpan data risiko, sistem langsung:

- Mengalikan nilai likelihood dan impact
- Menghasilkan skor risiko
- Mengklasifikasikan level risiko berdasarkan range tertentu

Contoh klasifikasi:

- 1–5 : Low Risk
- 6–12 : Medium Risk
- 12 : High Risk

Dengan otomatisasi ini:

- Mengurangi human error
- Menjaga konsistensi penilaian
- Mempercepat proses analisis

5. **Implementasi Role-Based Access Control (RBAC)**

Keamanan sistem menjadi aspek penting dalam implementasi. Oleh karena itu diterapkan Role-Based Access Control (RBAC).

Terdapat tiga peran utama:

- Admin
- Risk Officer
- User

Middleware digunakan untuk membatasi akses fitur berdasarkan role. Contohnya:

- Hanya Admin yang dapat mengelola data user.
- Risk Officer dapat menambah dan mengedit risiko.
- User hanya dapat melihat data sesuai izin.

Dengan penerapan RBAC, sistem menjadi lebih aman dan sesuai dengan prinsip least privilege.

6. **Implementasi Integrasi Artificial Intelligence (API LLM)**

Bagian paling inovatif dalam tahap implementasi adalah integrasi Large Language Model (LLM) melalui API untuk menghasilkan justifikasi risiko dan rekomendasi mitigasi secara otomatis.

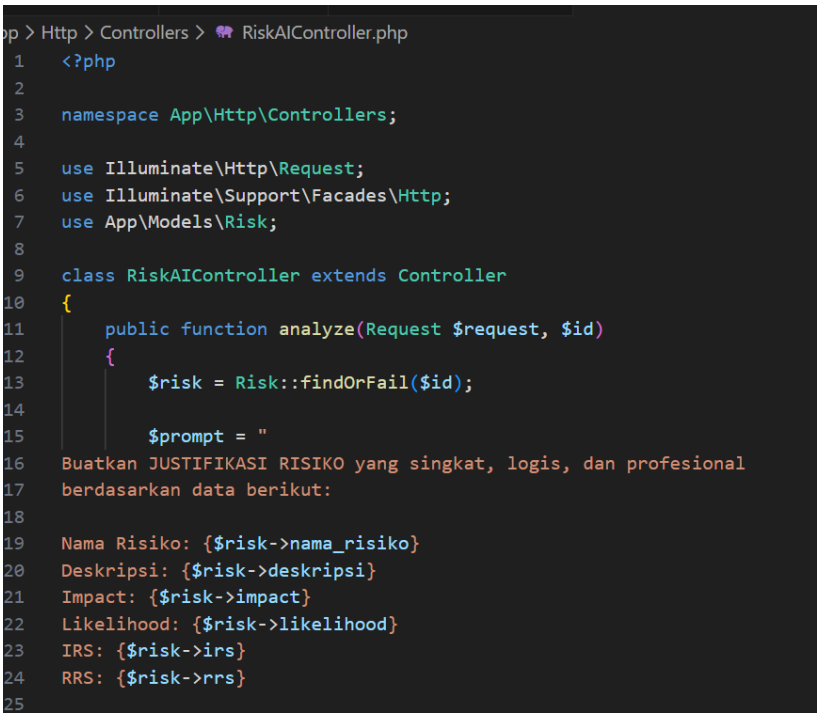
Sebelumnya, penyusunan justifikasi dilakukan secara manual sehingga:

- Membutuhkan waktu lama
- Kualitas narasi tidak konsisten
- Bergantung pada kemampuan individu

Dengan integrasi API AI, proses ini diotomatisasi melalui alur berikut:

1. User menginput data risiko (deskripsi, kategori, likelihood, impact).
2. Sistem menghitung skor dan level risiko.
3. Backend membentuk prompt terstruktur.
4. Sistem mengirim request ke API LLM.
5. API memproses dan mengembalikan output narasi.
6. Hasil ditampilkan dan disimpan dalam database.

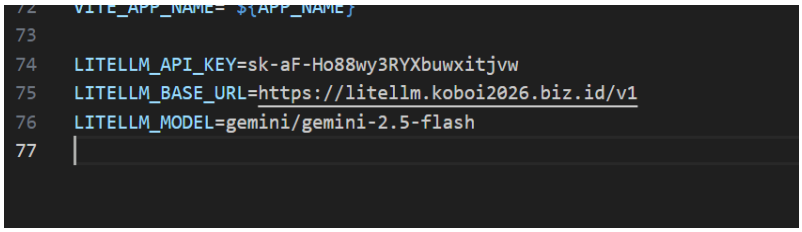
Contoh implementasi kode integrasi API:



Gambar 3. 30 Code RiskAIController.php

Pada implementasi integrasi Artificial Intelligence, sistem menggunakan **controller khusus** (RiskAIController.php) untuk memproses data risiko yang diinput pengguna dan mengirimkannya ke API LLM. Pada gambar pertama, terlihat bahwa controller membuat **prompt dinamis** berdasarkan data risiko yang tersimpan di database, termasuk nama risiko,

deskripsi, impact, likelihood, IRS, dan RRS. Prompt ini dirancang agar AI menghasilkan justifikasi risiko yang singkat, logis, dan profesional.



Gambar 3. 31 Code env. API Key

Selanjutnya, pada gambar 25 , terlihat pengaturan **API Key dan konfigurasi model LLM** di file .env.

- LITELLM_API_KEY digunakan sebagai autentikasi agar sistem dapat mengakses layanan AI.
- LITELLM_BASE_URL menunjukkan endpoint API yang akan menerima request.
- LITELLM_MODEL menentukan model LLM yang digunakan (gemini-2.5-flash) untuk menghasilkan narasi risiko.

Dengan pengaturan ini, backend Laravel dapat mengirim **HTTP request** ke API LLM menggunakan data risiko sebagai prompt, menerima respons berupa justifikasi dan rekomendasi mitigasi, lalu menyimpannya ke database untuk ditampilkan di frontend. Sistem ini menjadikan proses analisis risiko otomatis, cepat, dan konsisten, mengurangi ketergantungan pada penilaian manual.

Keamanan API dijaga dengan:

- Penyimpanan API key di file .env
- Tidak menuliskan API key secara langsung dalam kode
- Validasi response sebelum disimpan

Manfaat implementasi AI ini antara lain:

- Mempercepat penyusunan justifikasi
- Meningkatkan konsistensi kualitas analisis
- Mendukung pengambilan keputusan berbasis data
- Mengurangi ketergantungan pada proses manual

Arsitektur Integrasi Artificial Intelligence – Large Language Model (AI-LLM) :

Arsitektur integrasi Artificial Intelligence berbasis Large Language Model (LLM) pada sistem Intelligence Risk Register dirancang untuk mendukung proses penyusunan justifikasi dan rekomendasi mitigasi risiko secara otomatis. Integrasi ini memanfaatkan layanan Gemini AI API yang diakses melalui mekanisme HTTP request dari sistem berbasis Laravel.

Integrasi AI dilakukan tanpa menggantikan mekanisme perhitungan skor risiko utama. Perhitungan nilai risiko tetap dilakukan oleh sistem menggunakan rumus matematis Likelihood × Impact, sedangkan AI berfungsi sebagai pendukung analisis dalam menghasilkan narasi justifikasi dan rekomendasi mitigasi secara terstruktur.

Adapun alur arsitektur integrasi AI-LLM dalam sistem adalah sebagai berikut:

1. Pengguna Menginput Data Risiko

   Pengguna (Admin atau CRM) melakukan input data risiko melalui form yang tersedia pada sistem, meliputi nama risiko, aset, threat, vulnerability, impact, dan likelihood.

2. Perhitungan Nilai Risiko oleh Sistem

   Setelah data risiko disimpan, sistem secara otomatis menghitung nilai risiko menggunakan rumus:

   Risk Score=Likelihood×ImpactRisk\ Score = Likelihood \times

   ImpactRisk Score=Likelihood×Impact

   Hasil perhitungan ini digunakan untuk menentukan tingkat risiko (rendah, sedang, tinggi), namun belum menghasilkan justifikasi naratif.

3. Pembentukan Prompt Terstruktur

   Sistem kemudian membentuk prompt terstruktur berdasarkan data risiko yang telah diinput. Prompt tersebut memuat informasi sebagai berikut:

   - id (Primary Key)
   - tanggal_input
   - project
   - nama_risiko
   - aset
   - jenis_aset
   - threat
   - vulnerability
   - risk_event
   - risk_domain
   - category
   - risk_owner
   - control_owner
   - evidence
   - komitmen
   - impact

- likelihood
- rrs
- irs
- status
- justifikasi

Prompt dirancang dalam format instruksi yang jelas agar model LLM dapat menghasilkan justifikasi risiko dan rekomendasi mitigasi yang relevan serta kontekstual.

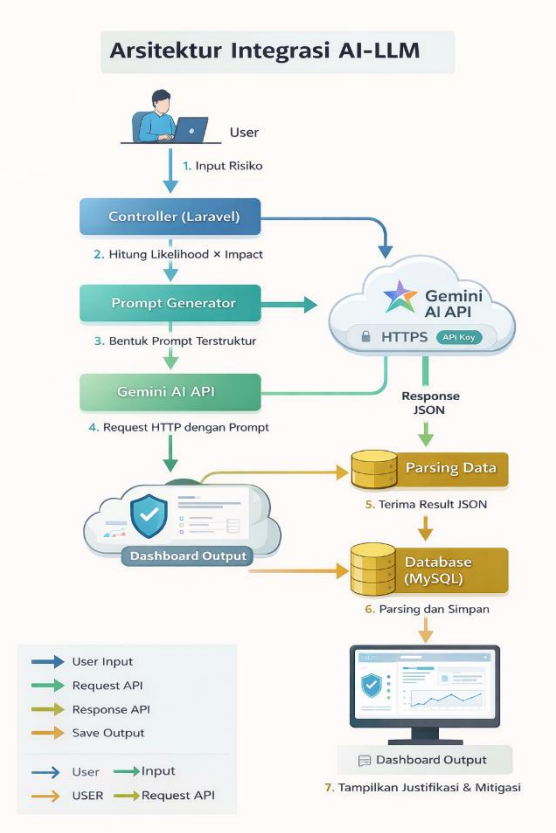
4. Pengiriman Request ke Gemini AI API

   Prompt yang telah terbentuk dikirimkan ke layanan Gemini AI melalui HTTP request menggunakan API key yang telah dikonfigurasi pada sistem Laravel. Proses komunikasi dilakukan secara secure melalui protokol HTTPS.
5. Penerimaan Response dalam Format JSON
6. Gemini AI API memproses prompt dan mengembalikan response dalam format JSON yang berisi hasil analisis berupa:
   - Justifikasi risiko
   - Rekomendasi mitigasi
7. Proses Parsing Response

   Sistem melakukan parsing terhadap response JSON untuk mengambil bagian teks yang relevan, kemudian memisahkan antara hasil justifikasi dan rekomendasi mitigasi.
8. Penyimpanan ke Database

   Hasil parsing kemudian disimpan ke dalam tabel RiskAnalysis dan Mitigation yang terhubung dengan entitas Risks pada basis data MySQL.
9. Penampilan Output pada Dashboard

   Justifikasi dan rekomendasi mitigasi yang telah tersimpan ditampilkan pada dashboard pengguna sesuai dengan hak akses masing-masing melalui mekanisme Role-Based Access Control (RBAC).



Gambar 3. 32 diagram alur integrasi AI (AI-LLM Flow Architecture)

7. **Pengujian Sistem (Testing)**

Setelah seluruh fitur selesai diimplementasikan, dilakukan pengujian menggunakan metode Black Box Testing untuk memastikan:

- Fungsi CRUD berjalan normal
- Perhitungan risiko akurat
- RBAC bekerja sesuai hak akses
- API AI menghasilkan output relevan

Setiap modul diuji secara terpisah (unit testing sederhana) dan kemudian diuji secara terintegrasi (integration testing).

3.6 Rencana Jadwal Penelitian (6 Bulan)

Rencana jadwal penelitian ini disusun berdasarkan tahapan **Software Development Life Cycle (SDLC) model Waterfall**, yang dilakukan secara berurutan dan sistematis. Setiap tahapan diselesaikan terlebih dahulu sebelum melanjutkan ke tahapan berikutnya, sesuai dengan karakteristik metode Waterfall sebagaimana dijelaskan oleh Pressman (2020).

Jadwal penelitian mencakup tahapan identifikasi masalah dan studi literatur, analisis kebutuhan sistem, perancangan sistem, implementasi dan pengujian sistem, serta evaluasi dan

penyusunan laporan akhir. Penyusunan jadwal ini bertujuan untuk memastikan seluruh proses penelitian berjalan terencana, terstruktur, dan sesuai dengan target waktu yang telah ditentukan.

Tabel 3. 2 Rencana Jadwal Penelitian

No	Kegiatan Penelitian	Bulan ke-1	Bulan ke-2	Bulan ke-3	Bulan ke-4	Bulan ke-5
1.	Identifikasi Masalah dan Studi Literatur	☑	☑			
2.	Analisis Kebutuhan Sistem		☑	☑		
3.	Perancangan Sistem (Database & UI)			☑	☑	
4.	Implementasi Sistem Website Risk Register dan Pengujian Sistem				☑	☑
5.	Evaluasi dan Penyusunan Laporan Akhir					☑