

**Pembangunan Manajemen Keuangan (Moneygement)**

Disusun Untuk Memenuhi Tugas Mata Kuliah Rekayasa Perangkat Lunak II

Dosen : Alif Finandita, S.Kom., M.T.

**KELAS IF-1**



Disusun oleh:

Dandy Muhamad Fadhillah	10123008
Reyhan Tahira	10123010
Dhaffa Khairu M. R. T.	10123013
Cherryl Jessica Oktavia	10123019
Rikza Danan Irdian	10123040

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNIK DAN ILMU KOMPUTER  
UNIVERSITAS KOMPUTER INDONESIA  
TAHUN 2025/2026**

## **1. Pendahuluan**

- Latar Belakang Sistem**

Banyak orang menghadapi kesulitan dalam menahan pengeluaran konsumtif dan tidak memiliki kebiasaan mencatat transaksi pribadi. Kurangnya kesadaran finansial ini diperparah dengan anggapan bahwa mencatat keuangan adalah hal yang merepotkan. Moneygement dirancang sebagai asisten keuangan pribadi yang praktis untuk mengatasi masalah tersebut.

- Permasalahan yang Ingin Diselesaikan:**

1. Kebiasaan pengguna yang konsumtif tanpa pencatatan.
2. Rendahnya retensi penggunaan aplikasi serupa akibat antarmuka yang rumit.
3. Keengganan pengguna melakukan pencatatan manual secara rutin.
4. Kesulitan dalam memahami pola pengeluaran secara mandiri.

- Tujuan Pengembangan Sistem**

Aplikasi ini bertujuan membantu pengguna mencatat pemasukan dan pengeluaran secara *real-time*, mengatur anggaran bulanan, menganalisis pola pengeluaran melalui grafik, serta menyusun laporan keuangan pribadi secara otomatis.

## **2. Validasi Ide**

- Metode Validasi Ide**

Kelompok kami melakukan pengumpulan data primer melalui penyebaran kuesioner secara daring menggunakan platform Google Form dan mendapat 37 responden dengan beragam latar belakang.

- Deskripsi Target Pengguna**

Target utama adalah kelompok usia produktif, di mana mayoritas responden berada pada rentang 15-24 tahun (84,8%). Berdasarkan latar belakang pekerjaan, responden didominasi oleh Pelajar/Mahasiswa (75,8%).

- Ringkasan Hasil Validasi:**

- Kesulitan Mengelola Keuangan: Sebanyak 51,4% responden merasa berada di level menengah dalam kesulitan mengatur keuangan, dan 35,1% lainnya merasa sangat kesulitan (Skala 4 dan 5).

- Masalah Pengeluaran Tak Terlacak: Mayoritas responden (67,5%) sering merasa uang mereka habis tanpa diketahui tujuannya (Skala 4 dan 5).
- Kebiasaan Konsumtif: Sekitar 48,6% responden mengakui memiliki kebiasaan konsumtif yang tinggi dan jarang mencatat pengeluaran harian mereka.
- Kesadaran dan Kebutuhan Solusi: Sebanyak 81,1% responden menganggap kontrol terhadap pengeluaran pribadi sangatlah penting. Selain itu, 83,8% responden menyatakan bahwa aplikasi pencatatan otomatis akan sangat membantu mereka.

- **Kesimpulan Kelayakan Ide:**

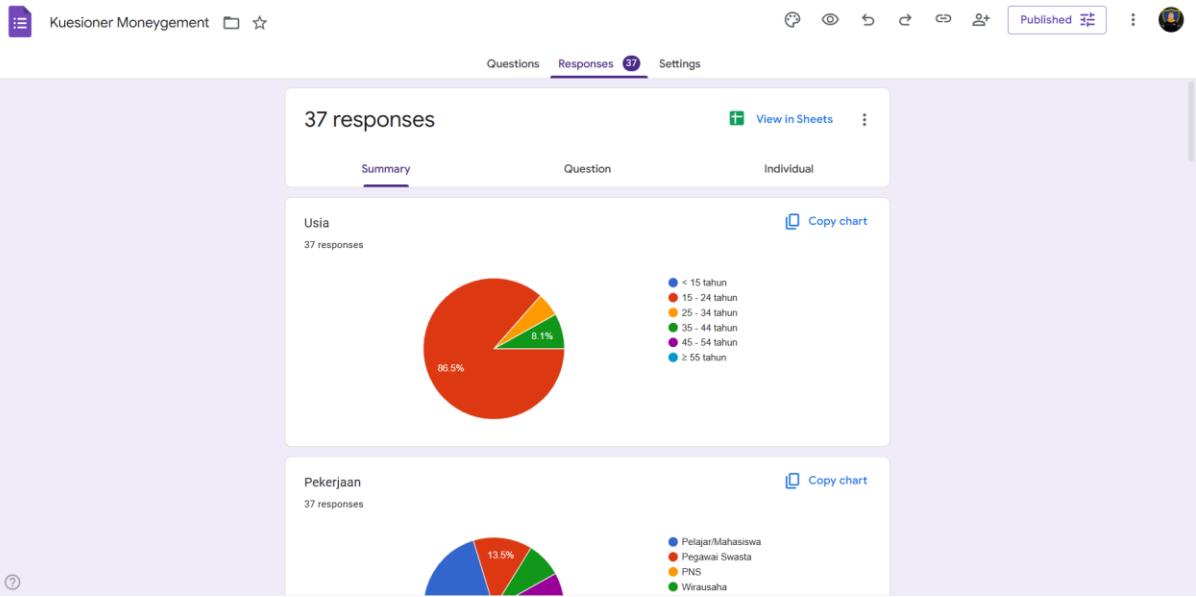
Berdasarkan hasil survei terhadap 37 responden, ide aplikasi Moneygement memiliki relevansi yang sangat tinggi karena mampu menjawab permasalahan nyata pengguna terkait "kebocoran finansial". Responden sangat membutuhkan fitur otomatisasi dan laporan berkala untuk mengatasi rasa malas dan lupa dalam mencatat keuangan manual.

Lebih dari 80% responden tertarik menggunakan aplikasi yang menawarkan pencatatan otomatis dan laporan keuangan

- **Lampiran Bukti Kuesioner**

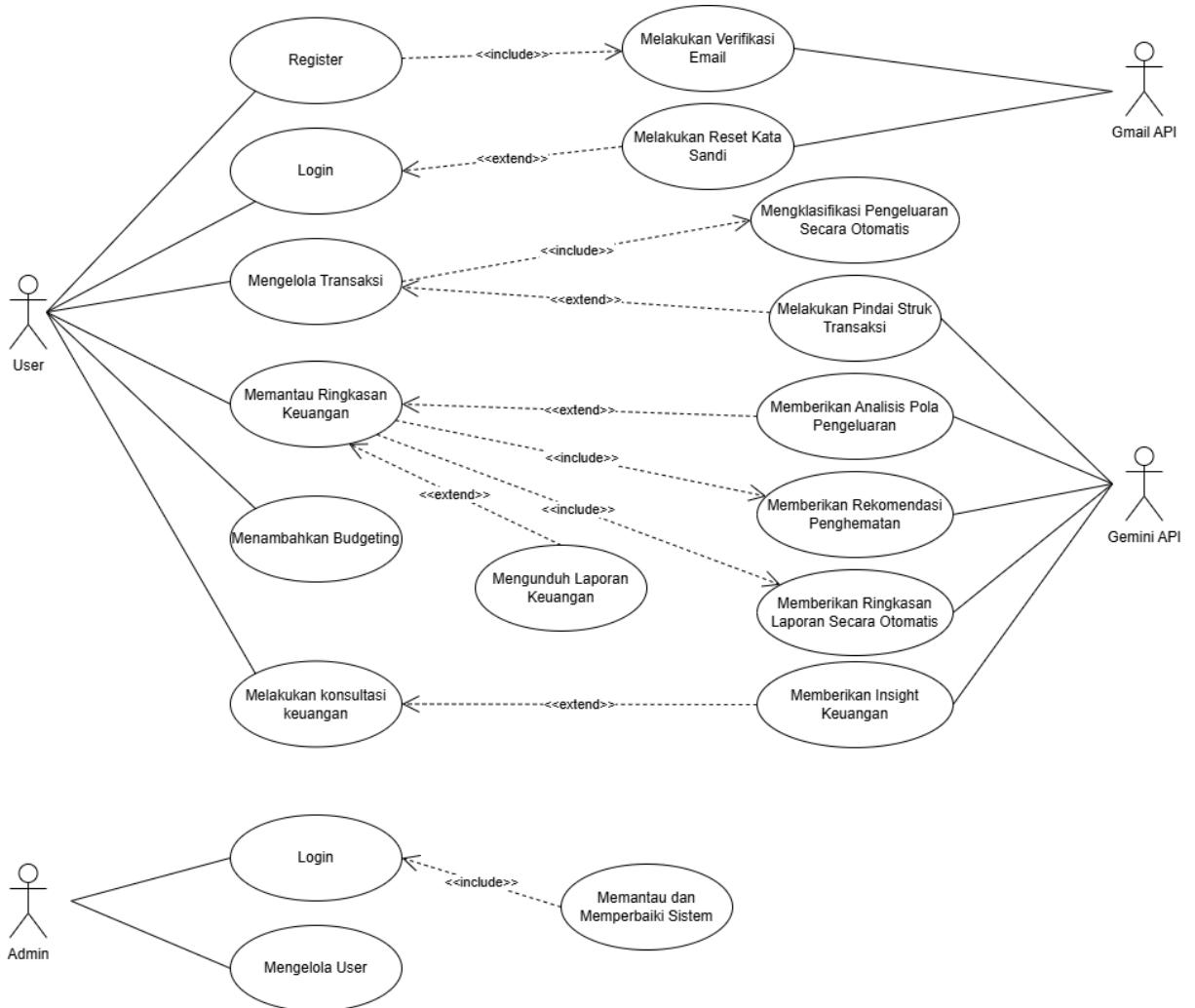
The screenshot shows a survey interface with the following details:

- Title:** Kuesioner Moneygement
- Description:** Kuesioner ini dibuat untuk memvalidasi asumsi awal (*risk assumptions*) terkait perilaku, kebutuhan, dan minat pengguna terhadap aplikasi Moneygement – sebuah aplikasi yang dirancang untuk membantu mengelola keuangan pribadi dengan lebih mudah dan teratur. Melalui kuesioner ini, kami ingin mengetahui apakah permasalahan yang diasumsikan benar dirasakan oleh pengguna, serta apakah solusi yang ditawarkan (Moneygement) dianggap relevan dan bermanfaat.
- Question:** Usia \*
- Options:**
  - < 15 tahun
  - 15 - 24 tahun
  - 25 - 34 tahun
  - 35 - 44 tahun
  - 45 - 54 tahun
  - ≥ 55 tahun
- Header Buttons:** Questions, Responses (37), Settings
- Header Icons:** Published, Share, Print, etc.



### 3. Use Case Diagram, Use Case Description, dan Diagram Activity

- Use Case Diagram



- Use Case Description

- Register

Use Case	Description
Description Detail	
Related Requirements	Memenuhi kebutuhan fungsional bagi User untuk memiliki akun sebelum dapat mengakses layanan aplikasi.
Goal In Context	Memberikan identitas digital kepada User agar data keuangan pribadinya dapat tersimpan dengan aman di dalam sistem.
Preconditions	User belum memiliki akun yang aktif dan sedang berada di halaman registrasi aplikasi.

<b>Successful End Condition</b>	Akun User berhasil dibuat, divalidasi, dan tersimpan dalam database dengan status aktif.
<b>Failed End Condition</b>	Sistem menampilkan pesan kesalahan dan akun tidak dibuat di dalam database.
<b>Primary Actors</b>	<b>User</b>
<b>Secondary Actors</b>	<b>Gmail API</b>
<b>Trigger</b>	<b>User</b> memilih fungsi "Register" atau "Daftar" pada antarmuka aplikasi.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. <b>User</b> memasukkan data identitas diri (Nama, Email, dan Password) pada form registrasi.</li> <li>2. <b>User</b> menekan tombol "Daftar" untuk mengirimkan data.</li> <li>3. Sistem memvalidasi bahwa semua field telah terisi dan format email sudah benar.</li> <li>4. Sistem mengecek database untuk memastikan email belum terdaftar.</li> <li>5. Sistem mengirimkan instruksi pengiriman email aktivasi kepada <b>Gmail API</b>.</li> <li>6. <b>Gmail API</b> mengirimkan email berisi tautan verifikasi ke kotak masuk User.</li> <li>7. <b>User</b> membuka email dan mengklik tautan verifikasi tersebut.</li> <li>8. Sistem mengaktifkan akun User dan menampilkan pesan sukses.</li> </ol>
<b>Extensions</b>	<p><b>3.1 Jika Form Tidak Lengkap:</b> Sistem memberikan peringatan dan meminta User mengisi kembali data yang kosong.</p> <p><b>4.1 Jika Email Sudah Terdaftar:</b> Sistem memberitahu User bahwa email tersebut sudah digunakan dan menghentikan proses.</p> <p><b>6.1 Jika Email Tidak Sampai:</b> User dapat menekan tombol "Kirim Ulang" untuk memicu kembali langkah nomor 5.</p>

- **Verifikasi Email**

<b>Use Case Description Detail</b>	<b>Description</b>
<b>Related Requirements</b>	Kebutuhan fungsional keamanan untuk memastikan validitas alamat email yang didaftarkan oleh User.

<b>Goal In Context</b>	Memastikan akun yang didaftarkan adalah valid dan milik pengguna yang sah sebelum diberikan akses penuh ke sistem.
<b>Preconditions</b>	User telah menyelesaikan pengisian form registrasi dan data awal telah tersimpan di database dengan status belum aktif.
<b>Successful End Condition</b>	Status akun User di database berubah menjadi aktif dan User dapat melakukan login ke sistem.
<b>Failed End Condition</b>	Akun tetap dalam status tidak aktif (terkunci) dan User tidak dapat mengakses fitur aplikasi.
<b>Primary Actors</b>	<b>User</b>
<b>Secondary Actors</b>	<b>Gmail API</b> (Sistem eksternal yang mengirimkan email verifikasi).
<b>Trigger</b>	Sistem menerima notifikasi sukses dari proses registrasi awal.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Sistem mengirimkan permintaan pengiriman tautan verifikasi ke <b>Gmail API</b>.</li> <li>2. <b>Gmail API</b> mengirimkan pesan berisi tautan unik ke alamat email <b>User</b>.</li> <li>3. <b>User</b> membuka pesan masuk pada aplikasi email mereka.</li> <li>4. <b>User</b> melakukan klik pada tautan verifikasi yang tersedia di dalam email.</li> <li>5. Sistem menerima instruksi dari tautan tersebut dan melakukan validasi token verifikasi.</li> <li>6. Sistem mengubah status akun <b>User</b> menjadi aktif di database.</li> <li>7. Sistem menampilkan pesan konfirmasi bahwa verifikasi berhasil.</li> </ol>
<b>Extensions</b>	<p><b>2.1 Email Tidak Sampai:</b> <b>User</b> dapat memilih opsi "Kirim Ulang" pada halaman aktivasi untuk memicu kembali langkah nomor 1.</p> <p><b>5.1 Tautan Kadaluwarsa:</b> Jika token sudah tidak valid, sistem menampilkan pesan error dan meminta <b>User</b> melakukan permintaan verifikasi baru.</p>

- o **Login User**

<b>Use Case</b>	<b>Description</b>
<b>Description Detail</b>	

<b>Related Requirements</b>	Kebutuhan fungsional keamanan untuk memastikan validitas alamat email yang didaftarkan oleh User.
<b>Goal In Context</b>	Memastikan akun yang didaftarkan adalah valid dan milik pengguna yang sah sebelum diberikan akses penuh ke sistem.
<b>Preconditions</b>	User telah menyelesaikan pengisian form registrasi dan data awal telah tersimpan di database dengan status belum aktif.
<b>Successful End Condition</b>	Status akun User di database berubah menjadi aktif dan User dapat melakukan login ke sistem.
<b>Failed End Condition</b>	Akun tetap dalam status tidak aktif (terkunci) dan User tidak dapat mengakses fitur aplikasi.
<b>Primary Actors</b>	<b>User</b>
<b>Secondary Actors</b>	-
<b>Trigger</b>	Sistem menerima notifikasi sukses dari proses registrasi awal.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. User memasukkan kredensial berupa alamat email dan kata sandi</li> <li>2. Sistem melakukan pencarian identitas pada tabel pengguna di database.</li> <li>3. Sistem memvalidasi kesesuaian antara kata sandi yang diinput dengan yang tersimpan (terenkripsi).</li> <li>4. Sistem memberikan akses masuk dan mengarahkan user ke fitur utama (Ringkasan Keuangan).</li> </ol>
<b>Extensions</b>	<ol style="list-style-type: none"> <li>2.1 Kredensial Salah : Sistem menampilkan pesan bahwa email atau kata sandi tidak cocok dan meminta input ulang.</li> <li>2.2 Lupa Kata Sandi : User memilih opsi pemulihan dan sistem mengarahkan ke use case “Melakukan Reset Kata Sandi”</li> </ol>

- **Mereset Katasandi**

<b>Use Case Detail</b>	<b>Description</b>
<b>Related Requirements</b>	Kebutuhan fitur keamanan agar User dapat memulihkan akses ke akun mereka saat kredensial login hilang atau terlupa.
<b>Goal In Context</b>	Memungkinkan User untuk memperbarui kata sandi lama dengan kata sandi baru melalui proses verifikasi identitas yang aman.

<b>Preconditions</b>	User berada pada kondisi tidak bisa mengakses akun dan memiliki akses ke alamat email yang sudah terdaftar di sistem.
<b>Successful End Condition</b>	Kata sandi baru berhasil diperbarui di database dan User dapat kembali melakukan login ke sistem.
<b>Failed End Condition</b>	Kata sandi gagal diperbarui karena email tidak ditemukan atau token verifikasi sudah kadaluwarsa.
<b>Primary Actors</b>	User.
<b>Secondary Actors</b>	Gmail API
<b>Trigger</b>	User melakukan permintaan pemulihan kata sandi pada sistem.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. User memasukkan alamat email yang terdaftar ke sistem.</li> <li>2. Sistem melakukan validasi keberadaan email di database.</li> <li>3. Sistem menginstruksikan Gmail API untuk mengirimkan tautan verifikasi reset kata sandi.</li> <li>4. <b>Gmail API</b> mengirimkan email berisi tautan verifikasi reset katandasdi ke kotak masuk user</li> <li>5. User mengakses tautan verifikasi yang dikirimkan ke email mereka.</li> <li>6. User memasukkan kata sandi baru dan konfirmasinya ke sistem.</li> <li>7. Sistem memvalidasi kesesuaian input kata sandi baru tersebut.</li> <li>8. Sistem memperbarui kata sandi di database dan memberikan konfirmasi sukses.</li> </ol>
<b>Extensions</b>	<p>2.1 Email Tidak Ditemukan: Sistem menginformasikan bahwa data tidak tersedia dan proses dihentikan.</p> <p>7.1 Input Tidak Cocok: Sistem meminta User melakukan pengulangan input kata sandi baru agar sinkron.</p>

- o Mengelola Transaksi

<b>Use Case Description Detail</b>	<b>Description</b>
<b>Related Requirements</b>	Kebutuhan fungsional untuk mencatat, melihat, mengubah, dan menghapus data pemasukan serta pengeluaran.
<b>Goal In Context</b>	Memberikan kontrol penuh kepada User atas catatan riwayat arus kas harian untuk memantau kondisi finansial.

<b>Preconditions</b>	User telah berhasil login ke dalam aplikasi dan berada pada halaman kelola transaksi.
<b>Successful End Condition</b>	Data transaksi berhasil diperbarui (ditambah/diubah/dihapus) dan tersimpan dengan benar di database.
<b>Failed End Condition</b>	Transaksi tidak tersimpan karena kesalahan input atau gangguan teknis, dan sistem menampilkan pesan error.
<b>Primary Actors</b>	User.
<b>Secondary Actors</b>	
<b>Trigger</b>	User memilih menu atau tombol untuk mengelola transaksi pada dashboard.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. User memilih aksi yang ingin dilakukan (Tambah, Edit, atau Hapus transaksi).</li> <li>2. User memasukkan atau memperbarui data transaksi (Nominal, Kategori, Tanggal, dan Keterangan).</li> <li>3. Sistem memvalidasi kelengkapan dan format data yang dimasukkan.</li> <li>4. Sistem menyimpan perubahan data ke dalam database.</li> <li>5. Sistem memperbarui tampilan daftar transaksi pada antarmuka User.</li> </ol>
<b>Extensions</b>	<p>3.1 Data Tidak Lengkap: Sistem memberikan peringatan kepada User untuk melengkapi field yang kosong.</p> <p>4.1 Kegagalan Database: Jika sistem gagal menyimpan data, sistem memberikan notifikasi kesalahan teknis.</p>

- **Pindai Struk Transaksi**

<b>Use Case Description Detail</b>	<b>Description</b>
<b>Related Requirements</b>	Kebutuhan teknologi otomasi untuk mengekstraksi data (nominal, tanggal, kategori) dari gambar fisik struk belanja.
<b>Goal In Context</b>	Mempercepat proses input data transaksi bagi <b>User</b> tanpa perlu melakukan pengetikan manual.
<b>Preconditions</b>	<b>User</b> telah login ke aplikasi dan memiliki foto struk yang jelas.

<b>Successful End Condition</b>	Data dari struk berhasil diekstraksi dan tersimpan sebagai transaksi baru dalam database.
<b>Failed End Condition</b>	Sistem gagal mengekstraksi data dari gambar dan menampilkan pesan kesalahan.
<b>Primary Actors</b>	<b>User.</b>
<b>Secondary Actors</b>	<b>Gemini API</b>
<b>Trigger</b>	<b>User</b> menekan ikon kamera atau tombol "Pindai Struk".
<b>Main Flow</b>	<ol style="list-style-type: none"> <li><b>User</b> mengambil foto struk fisik atau mengunggah gambar dari galeri.</li> <li>Sistem mengirimkan data gambar tersebut ke <b>Gemini API</b> untuk diproses.</li> <li><b>Gemini API</b> mengekstraksi informasi toko, nominal, tanggal, kategori transaksi, dan deskripsi.</li> <li>Sistem menampilkan hasil ekstraksi data kepada <b>User</b> untuk ditinjau.</li> <li><b>User</b> mengonfirmasi data dan menekan tombol "Simpan".</li> <li>Sistem menyimpan data transaksi ke dalam database.</li> </ol>
<b>Extensions</b>	<p><b>2.1 Gambar Buram:</b> Sistem memberikan peringatan bahwa gambar tidak jelas dan meminta User foto ulang.</p> <p><b>3.1 API Tidak Merespon:</b> Sistem memberikan notifikasi gangguan layanan dan menyarankan input manual.</p>

- **Mengklasifikasikan Pengeluaran Secara Otomatis**

<b>Use Case Detail</b>	<b>Description</b>
<b>Related Requirements</b>	Kebutuhan fitur cerdas untuk kategorisasi data secara <i>real-time</i> guna menjaga akurasi laporan keuangan.
<b>Goal In Context</b>	Memastikan setiap data transaksi (pengeluaran/pemasukan) memiliki label kategori yang relevan melalui analisis AI.
<b>Preconditions</b>	Sistem telah menerima data teks transaksi baru yang belum memiliki kategori.
<b>Successful End Condition</b>	Transaksi berhasil disimpan di database lengkap dengan kategori yang tepat hasil analisis AI.

<b>Failed End Condition</b>	Sistem gagal mengidentifikasi kategori dan memberikan label default "Lain-lain".
<b>Primary Actors</b>	<b>User.</b>
<b>Secondary Actors</b>	<b>Gemini API.</b>
<b>Trigger</b>	Masuknya data transaksi baru ke dalam sistem (dari input manual atau pindai).
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Sistem menangkap deskripsi teks dari transaksi baru.</li> <li>2. Sistem mengirimkan teks tersebut ke <b>Gemini API</b>.</li> <li>3. <b>Gemini API</b> melakukan analisis konteks untuk menentukan kategori yang sesuai.</li> <li>4. Sistem menerima hasil kategori dari API.</li> <li>5. Sistem memperbarui data transaksi dengan kategori tersebut di database.</li> </ol>
<b>Extensions</b>	<p><b>3.1 Masalah Koneksi/API:</b> Jika API tidak merespon, sistem menetapkan kategori "Lain-lain".</p> <p><b>4.1 Koreksi Manual:</b> User mengubah kategori secara manual jika hasil otomatis dirasa tidak akurat.</p>

- **Memantau Ringkasan Keuangan**

<b>Use Case Detail</b>	<b>Description</b>
<b>Use Case Name</b>	<b>Memantau Ringkasan Keuangan</b>
<b>Related Requirements</b>	Kebutuhan fitur untuk menyajikan visualisasi data transaksi agar User dapat memahami kondisi finansial dengan cepat dan akurat.
<b>Goal In Context</b>	Memberikan pemahaman menyeluruh mengenai kondisi keuangan (pemasukan vs pengeluaran) melalui representasi data visual.
<b>Preconditions</b>	User telah login dan memiliki data transaksi yang tersimpan di dalam database.
<b>Successful End Condition</b>	Sistem menyajikan visualisasi statistik (grafik) yang akurat berdasarkan data transaksi terbaru milik User.
<b>Failed End Condition</b>	Grafik gagal dimuat atau menampilkan informasi kosong karena tidak adanya data transaksi yang tersedia.
<b>Primary Actors</b>	<b>User.</b>
<b>Secondary Actors</b>	-

<b>Trigger</b>	User memilih fitur atau menu Ringkasan Keuangan pada aplikasi.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. <b>User</b> memilih fitur Ringkasan Keuangan.</li> <li>2. Sistem melakukan pengambilan data seluruh transaksi User dari database.</li> <li>3. Sistem melakukan kalkulasi agregat (penjumlahan total) berdasarkan kategori dan periode waktu.</li> <li>4. Sistem memetakan hasil perhitungan ke dalam format visualisasi grafik (seperti <i>Pie Chart</i> atau <i>Bar Chart</i>).</li> <li>5. Sistem menyajikan informasi ringkasan tersebut kepada <b>User</b>.</li> </ol>
<b>Extensions</b>	<p><b>2.1 Data Tidak Tersedia:</b> Sistem memberikan informasi bahwa belum ada data yang dapat diringkas dan menyarankan User untuk menambahkan transaksi baru.</p>

- **Memberikan Analisis Pola Pengeluaran**

Use Case Detail	Description
<b>Use Case Name</b>	<b>Memberikan Analisis Pola Keuangan</b>
<b>Related Requirements</b>	Kebutuhan fitur cerdas untuk mengidentifikasi kebiasaan transaksi User agar dapat memberikan saran finansial yang tepat.
<b>Goal In Context</b>	Menyajikan temuan mendalam mengenai perilaku pengeluaran dan pemasukan User dalam periode tertentu menggunakan bantuan AI.
<b>Preconditions</b>	User telah memiliki riwayat transaksi yang cukup di database dan sedang menggunakan fitur konsultasi atau ringkasan.
<b>Successful End Condition</b>	User mendapatkan laporan naratif mengenai pola pengeluarannya (misal: "Pengeluaran kopi meningkat di akhir pekan").
<b>Failed End Condition</b>	Sistem tidak dapat menemukan pola yang konsisten atau terjadi kegagalan pada layanan AI.
<b>Primary Actors</b>	<b>User</b> .
<b>Secondary Actors</b>	<b>Gemini API</b>

<b>Trigger</b>	User meminta analisis perilaku keuangan atau sistem mendeteksi tren yang signifikan.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. User memilih fitur Analisis pola pengeluaran.</li> <li>2. Sistem mengumpulkan data riwayat transaksi User (deskripsi, kategori, waktu, dan nominal).</li> <li>3. Sistem meneruskan kumpulan data tersebut ke <b>Gemini API</b>.</li> <li>4. <b>Gemini API</b> melakukan analisis statistik dan mendeteksi pola yang berulang atau tidak wajar.</li> <li>5. <b>Gemini API</b> merumuskan hasil analisis dalam bentuk teks penjelasan.</li> <li>6. Sistem menyajikan hasil analisis pola tersebut kepada <b>User</b>.</li> </ol>
<b>Extensions</b>	<p><b>4.1 Data Tidak Mencukupi:</b> Jika data terlalu sedikit, sistem memberikan informasi bahwa pola belum dapat terbentuk.</p>

- **Memberikan Rekomendasi Penghematan**

Use Case Detail	Description
<b>Use Case Name</b>	<b>Memberikan Rekomendasi Penghematan</b>
<b>Related Requirements</b>	Kebutuhan fitur proaktif yang membantu User mengurangi pengeluaran berdasarkan hasil analisis data transaksi.
<b>Goal In Context</b>	Menyediakan saran langkah nyata yang dapat dilakukan User untuk mengoptimalkan keuangan mereka.
<b>Preconditions</b>	Sistem telah memiliki data transaksi yang cukup dan telah melakukan analisis pola pengeluaran.
<b>Successful End Condition</b>	User menerima saran penghematan yang spesifik dan relevan (misal: "Kurangi pengeluaran kategori Hiburan sebesar 10%").
<b>Failed End Condition</b>	Sistem gagal menghasilkan rekomendasi karena data tidak mencukupi atau gangguan pada layanan AI.
<b>Primary Actors</b>	User.
<b>Secondary Actors</b>	<b>Gemini API</b>
<b>Trigger</b>	User menanyakan saran penghematan atau sistem memberikan notifikasi saran secara berkala.

<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. User memilih fitur Rekomendasi penghematan.</li> <li>2. Sistem mengidentifikasi kategori pengeluaran yang melebihi rata-rata atau anggaran.</li> <li>3. Sistem meneruskan data temuan tersebut ke <b>Gemini API</b>.</li> <li>4. <b>Gemini API</b> menyusun beberapa opsi strategi penghematan yang realistik.</li> <li>5. Sistem menyajikan poin-poin rekomendasi penghematan tersebut kepada <b>User</b>.</li> </ol>
<b>Extensions</b>	<p><b>4.1 Strategi Tidak Ditemukan:</b> Jika pengeluaran sudah optimal, sistem memberikan apresiasi dan saran untuk mempertahankan pola tersebut.</p>

- **Memberikan Ringkasan Laporan Secara Otomatis**

<b>Use Case Detail</b>	<b>Description</b>
<b>Use Case Name</b>	<b>Memberikan Ringkasan Laporan Secara Otomatis</b>
<b>Related Requirements</b>	Kebutuhan untuk menyederhanakan data transaksi yang kompleks menjadi poin-poin informasi yang mudah dipahami <b>User</b> .
<b>Goal In Context</b>	Menyajikan rangkuman naratif mengenai aktivitas keuangan <b>User</b> (misal: mingguan atau bulanan) secara instan.
<b>Preconditions</b>	<b>User</b> memiliki histori transaksi dalam periode waktu tertentu dan sistem terhubung dengan layanan AI.
<b>Successful End Condition</b>	<b>User</b> menerima teks ringkasan yang mencakup total pengeluaran, kategori terbesar, dan perbandingan dengan periode sebelumnya.
<b>Failed End Condition</b>	Sistem hanya menampilkan daftar transaksi tanpa ringkasan karena kegagalan API atau data tidak mencukupi.
<b>Primary Actors</b>	<b>User</b> .
<b>Secondary Actors</b>	<b>Gemini API</b>
<b>Trigger</b>	<b>User</b> membuka fitur laporan atau meminta sistem untuk membuat ringkasan periode tertentu.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. <b>User</b> memilih fitur ringkasan laporan secara otomatis.</li> <li>2. Sistem mengumpulkan data agregat transaksi (total, kategori, frekuensi) dalam periode yang dipilih.</li> <li>3. Sistem meneruskan data agregat tersebut ke <b>Gemini API</b>.</li> </ol>

	<p>4. <b>Gemini API</b> mengolah data tersebut menjadi kalimat ringkasan yang informatif.</p> <p>5. Sistem menyajikan teks ringkasan tersebut pada bagian atas fitur laporan User.</p>
<b>Extensions</b>	<b>4.1 API Gagal:</b> Sistem menampilkan pesan bahwa ringkasan otomatis tidak tersedia untuk sementara waktu.

- **Mengunduh Laporan Keuangan**

Use Case Detail	Description
<b>Use Case Name</b>	<b>Mengunduh Laporan Keuangan</b>
<b>Related Requirements</b>	Kebutuhan fitur agar User dapat menyimpan dokumen riwayat keuangan secara luring ( <i>offline</i> ) untuk keperluan dokumentasi atau cetak.
<b>Goal In Context</b>	Menghasilkan berkas digital (seperti PDF atau Excel) yang berisi rekapitulasi transaksi User dalam periode tertentu.
<b>Preconditions</b>	User memiliki data transaksi yang tersimpan dan telah memilih periode laporan yang diinginkan.
<b>Successful End Condition</b>	Berkas laporan berhasil dibuat oleh sistem dan terunduh ke perangkat penyimpanan lokal User.
<b>Failed End Condition</b>	Berkas gagal dibuat karena kesalahan sistem atau User tidak memiliki riwayat transaksi pada periode tersebut.
<b>Primary Actors</b>	User.
<b>Secondary Actors</b>	-
<b>Trigger</b>	User memilih perintah atau opsi untuk mengunduh laporan pada sistem.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. User menentukan parameter periode laporan (misal: bulan Januari).</li> <li>2. Sistem melakukan agregasi data transaksi berdasarkan periode yang dipilih.</li> <li>3. Sistem menyusun data tersebut ke dalam format dokumen yang ditentukan (PDF/Excel).</li> <li>4. Sistem mengonversi data menjadi berkas digital.</li> </ol>

	5. Sistem mengirimkan berkas tersebut ke peramban atau perangkat User untuk diunduh.
<b>Extensions</b>	<p><b>2.1 Data Tidak Ditemukan:</b> Sistem memberikan informasi bahwa laporan tidak bisa dibuat karena tidak ada aktivitas transaksi di periode tersebut.</p> <p><b>4.1 Gagal Generate Berkas:</b> Sistem menginformasikan adanya kendala teknis dan meminta User mencoba kembali.</p>

- **Mengelola Budgeting**

<b>Use Case Detail</b>	<b>Description</b>
<b>Use Case Name</b>	<b>Mengelola Anggaran</b>
<b>Related Requirements</b>	Kebutuhan fitur agar User dapat merencanakan batas pengeluaran pada kategori tertentu untuk mengontrol kondisi finansial.
<b>Goal In Context</b>	Membantu User mengatur batasan limit saldo pada kategori pengeluaran tertentu agar tidak terjadi pemborosan.
<b>Preconditions</b>	User telah melakukan login dan memiliki kategori pengeluaran yang sudah terdaftar di sistem.
<b>Successful End Condition</b>	Data anggaran berhasil disimpan atau diperbarui di database dan sistem mulai melakukan pemantauan terhadap transaksi terkait.
<b>Failed End Condition</b>	Perubahan anggaran gagal disimpan karena kesalahan input (nominal negatif) atau gangguan database.
<b>Primary Actors</b>	<b>User.</b>
<b>Secondary Actors</b>	-
<b>Trigger</b>	<b>User</b> memilih fitur anggaran atau ingin menyesuaikan limit pengeluaran pada kategori tertentu.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. <b>User</b> memilih kategori pengeluaran yang ingin diberi anggaran (misal: "Makanan").</li> <li>2. <b>User</b> menentukan batas nominal anggaran dan periode waktunya.</li> <li>3. Sistem melakukan validasi terhadap input nominal yang dimasukkan.</li> <li>4. Sistem menyimpan data anggaran tersebut ke dalam database.</li> <li>5. Sistem memberikan konfirmasi bahwa anggaran telah berhasil ditetapkan.</li> </ol>

<b>Extensions</b>	<p><b>3.1 Nominal Tidak Valid:</b> Jika User memasukkan angka nol atau negatif, sistem menampilkan pesan error dan meminta input ulang.</p> <p><b>4.1 Anggaran Sudah Ada:</b> Jika kategori tersebut sudah memiliki anggaran, sistem menawarkan opsi untuk memperbarui atau membatalkan.</p>
-------------------	--

- **Melakukan Konsultasi Keuangan**

<b>Use Case Detail</b>	<b>Description</b>
<b>Use Case Name</b>	<b>Melakukan Konsultasi Keuangan</b>
<b>Related Requirements</b>	Kebutuhan fitur interaktif berbasis kecerdasan buatan untuk membantu User mendapatkan solusi atau jawaban terkait masalah finansial secara <i>real-time</i> .
<b>Goal In Context</b>	Membantu User memecahkan masalah atau mendapatkan informasi keuangan melalui percakapan dua arah dengan asisten cerdas.
<b>Preconditions</b>	User telah login ke dalam sistem dan memiliki koneksi internet untuk terhubung ke layanan eksternal.
<b>Successful End Condition</b>	User mendapatkan jawaban atau saran yang relevan yang ditampilkan di dalam jendela percakapan.
<b>Failed End Condition</b>	Sistem tidak memberikan respon atau menampilkan pesan kesalahan karena kegagalan pada layanan API.
<b>Primary Actors</b>	<b>User.</b>
<b>Secondary Actors</b>	<b>Gemini API .</b>
<b>Trigger</b>	User mengirimkan pesan atau pertanyaan pertama pada fitur konsultasi.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. User memasukkan pertanyaan atau keluhan keuangan pada kolom input.</li> <li>2. Sistem meneruskan pesan tersebut ke <b>Gemini API</b> beserta konteks transaksi User yang relevan.</li> <li>3. <b>Gemini API</b> memproses pesan tersebut dan merumuskan jawaban yang sesuai.</li> <li>4. Sistem menerima teks jawaban dari <b>Gemini API</b>.</li> </ol>

	5. Sistem menyajikan jawaban tersebut pada antarmuka percakapan <b>User</b> .
<b>Extensions</b>	<p><b>3.1 Masalah Koneksi:</b> Jika layanan AI tidak merespon, sistem memberikan informasi bahwa layanan sedang sibuk dan meminta User mencoba lagi nanti.</p> <p><b>4.1 Pesan Tidak Dimengerti:</b> Jika input terlalu ambigu, sistem meminta User untuk memperjelas pertanyaannya.</p>

- **Memberikan Insight Keuangan**

<b>Use Case Detail</b>	<b>Description</b>
<b>Use Case Name</b>	<b>Memberikan Insight Keuangan</b>
<b>Related Requirements</b>	Kebutuhan fitur proaktif untuk memberikan informasi mendalam (pengetahuan baru) mengenai kondisi finansial yang tidak terlihat secara kasat mata oleh User.
<b>Goal In Context</b>	Menyajikan poin-poin informasi strategis mengenai kondisi keuangan User berdasarkan analisis data historis.
<b>Preconditions</b>	User telah memiliki data transaksi yang mencukupi untuk dianalisis oleh sistem.
<b>Successful End Condition</b>	User menerima informasi berharga (seperti: "Pengeluaran bulan ini lebih rendah 20% dari biasanya") yang dapat membantu pengambilan keputusan.
<b>Failed End Condition</b>	Sistem gagal memberikan insight karena ketiadaan data atau kegagalan pada mesin pengolah data (AI).
<b>Primary Actors</b>	<b>User</b>
<b>Secondary Actors</b>	<b>Gemini API</b>
<b>Trigger</b>	User mengakses fitur insight atau sistem mendeteksi adanya informasi penting untuk ditampilkan.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. User memilih fitur insight keuangan.</li> <li>2. Sistem melakukan ekstraksi data transaksi dan kategori milik <b>User</b>.</li> <li>3. Sistem mengirimkan dataset tersebut ke <b>Gemini API</b>.</li> <li>4. <b>Gemini API</b> melakukan komparasi data (perbandingan antar periode atau target).</li> </ol>

	<p>5. <b>Gemini API</b> menghasilkan teks insight yang bersifat informatif dan edukatif.</p> <p>6. Sistem menyajikan poin-poin insight tersebut kepada <b>User</b>.</p>
<b>Extensions</b>	<p><b>4.1 Data Minim:</b> Jika data belum mencukupi untuk dibandingkan, sistem menginformasikan bahwa insight akan tersedia setelah lebih banyak transaksi tercatat.</p> <p><b>5.1 Kegagalan API:</b> Sistem menampilkan pesan bahwa analisis otomatis sedang tidak tersedia.</p>

- o **Login Admin**

Use Case Detail	Description
<b>Use Case Name</b>	<b>Melakukan Autentikasi Admin</b>
<b>Related Requirements</b>	
<b>Goal In Context</b>	Memastikan hanya personil berwenang yang dapat mengakses fungsi pengelolaan tingkat tinggi.
<b>Preconditions</b>	
<b>Successful End Condition</b>	Admin berhasil masuk ke dalam sistem dengan hak akses penuh.
<b>Failed End Condition</b>	
<b>Primary Actors</b>	<b>Admin</b>
<b>Secondary Actors</b>	
<b>Trigger</b>	Admin mengakses gerbang masuk manajemen sistem.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. <b>Admin</b> memasukkan kredensial (username dan password).</li> <li>2. Sistem memvalidasi data terhadap tabel administrator di database.</li> <li>3. Sistem memberikan akses ke fungsi manajemen sistem.</li> </ol>
<b>Extensions</b>	<p><b>2.1 Kredensial Salah:</b> Sistem menolak akses dan meminta input ulang.</p>

- **Memantau dan Memperbaiki Sistem**

<b>Use Case Detail</b>	<b>Description</b>
<b>Use Case Name</b>	<b>Memantau dan Memperbaiki Sistem</b>
<b>Related Requirements</b>	
<b>Goal In Context</b>	Menjaga stabilitas aplikasi dengan mengawasi kesehatan server dan integritas data internal secara rutin.
<b>Preconditions</b>	<b>Admin</b> telah berhasil melakukan autentikasi ke dalam sistem.
<b>Successful End Condition</b>	Sistem berjalan stabil dan setiap anomali data atau performa berhasil dinormalkan kembali.
<b>Failed End Condition</b>	
<b>Primary Actors</b>	<b>Admin</b>
<b>Secondary Actors</b>	
<b>Trigger</b>	<b>Admin</b> melakukan pengecekan rutin pada sistem atau munculnya laporan <i>error</i> internal.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. <b>Admin</b> mengakses log aktivitas sistem dan statistik penggunaan sumber daya server.</li> <li>2. Sistem menyajikan informasi kinerja basis data dan lalu lintas data internal.</li> <li>3. <b>Admin</b> melakukan tindakan perbaikan (seperti optimasi database atau pembersihan file sampah).</li> <li>4. Sistem memperbarui status kesehatan sistem dan mencatat aktivitas perbaikan.</li> </ol>
<b>Extensions</b>	<b>3.1 Kegagalan Proses:</b> Sistem memberikan peringatan jika tindakan perbaikan gagal dieksekusi karena masalah izin akses ( <i>permission</i> ).

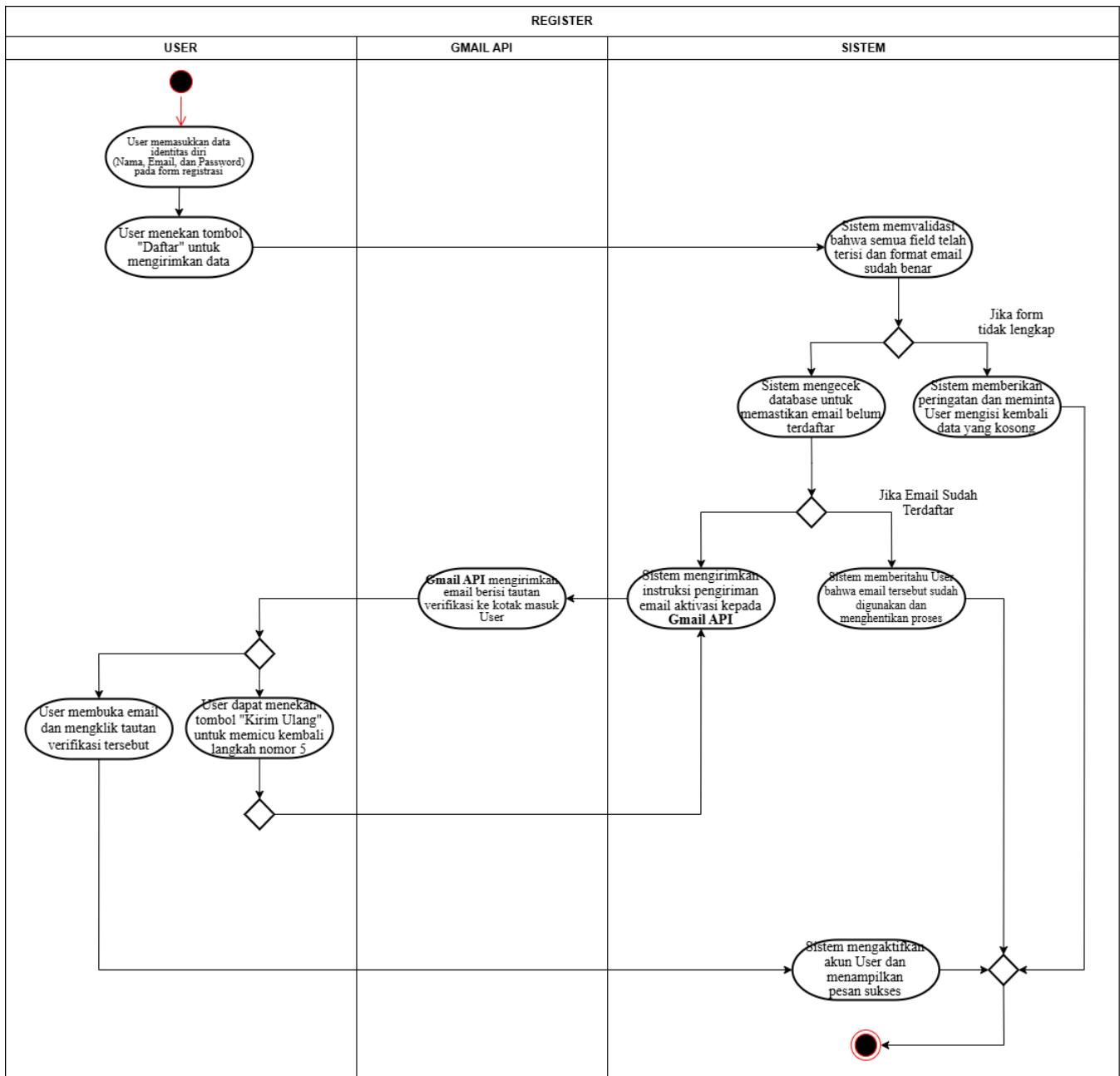
- **Mengelola User**

<b>Use Case Detail</b>	<b>Description</b>
<b>Use Case Name</b>	<b>Mengelola Pengguna</b>
<b>Related Requirements</b>	

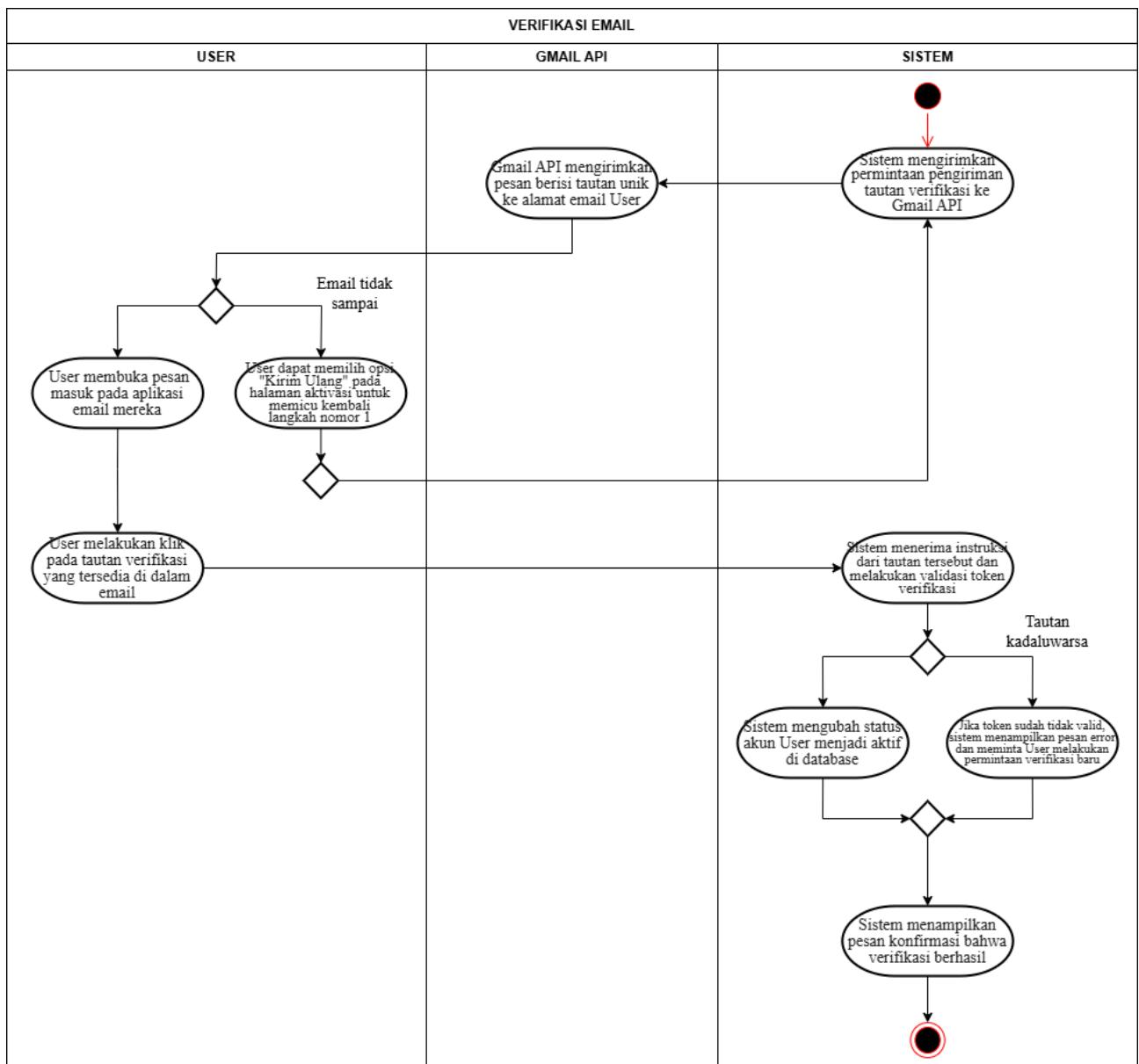
<b>Goal In Context</b>	Melakukan kontrol penuh terhadap akun pengguna yang terdaftar di dalam aplikasi.
<b>Preconditions</b>	Admin telah melakukan autentikasi ke dalam sistem.
<b>Successful End Condition</b>	Data pengguna berhasil diperbarui, dihapus, atau disesuaikan statusnya di database.
<b>Failed End Condition</b>	
<b>Primary Actors</b>	<b>Admin</b>
<b>Secondary Actors</b>	
<b>Trigger</b>	Admin perlu melakukan tindakan moderasi terhadap salah satu atau kelompok akun pengguna.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li><b>Admin</b> melihat daftar seluruh akun pengguna yang terdaftar.</li> <li><b>Admin</b> menentukan aksi pengelolaan (seperti blokir, hapus, atau edit status).</li> <li>Sistem memproses instruksi tersebut ke dalam database.</li> <li>Sistem memberikan konfirmasi bahwa pengelolaan data pengguna berhasil dilakukan.</li> </ol>
<b>Extensions</b>	<b>2.1 Akun Tidak Ditemukan:</b> Sistem menginformasikan jika data pengguna yang dicari tidak tersedia.

- **Diagram Activity**

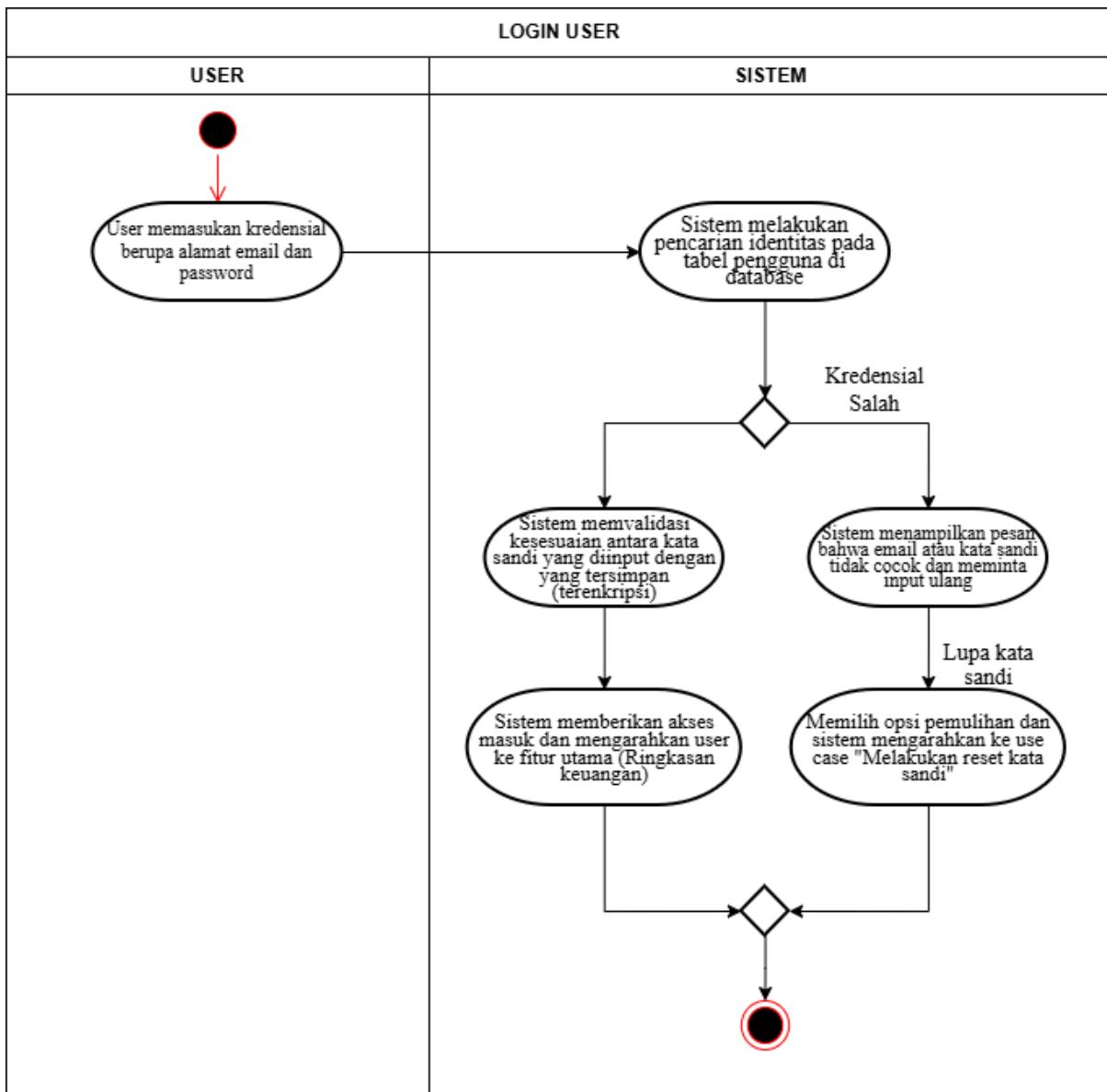
- **Register**



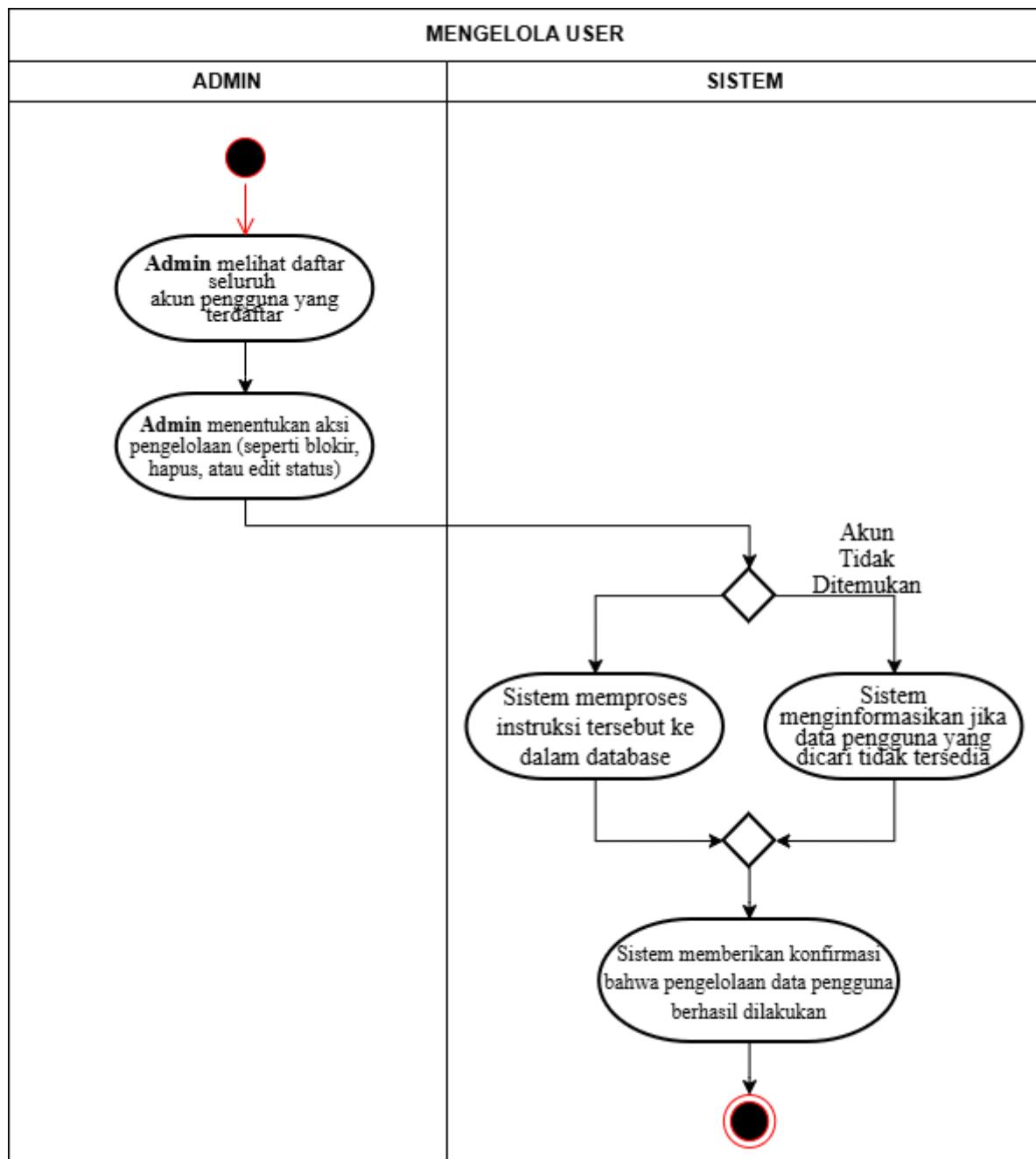
- Verifikasi Email



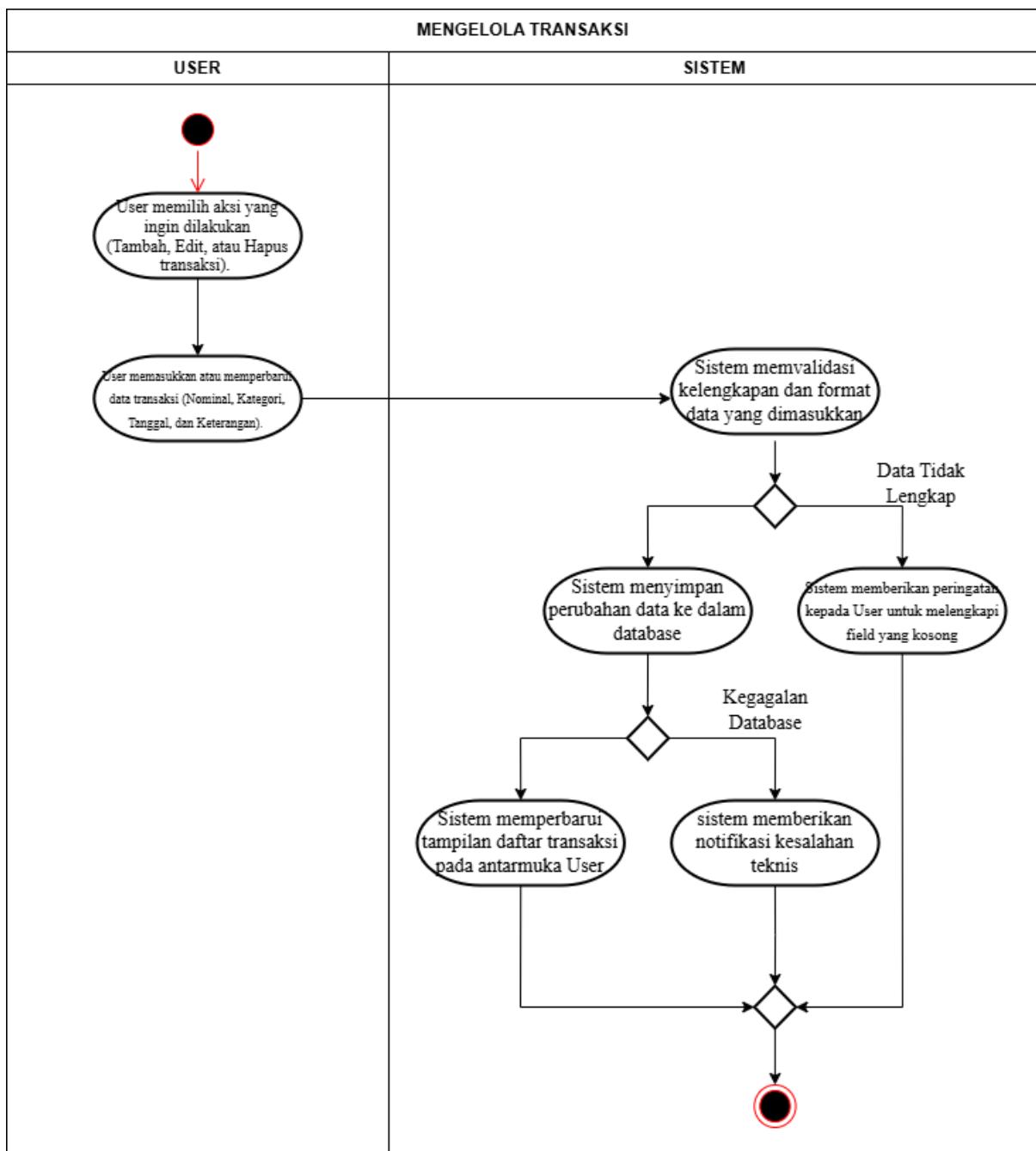
- Login User



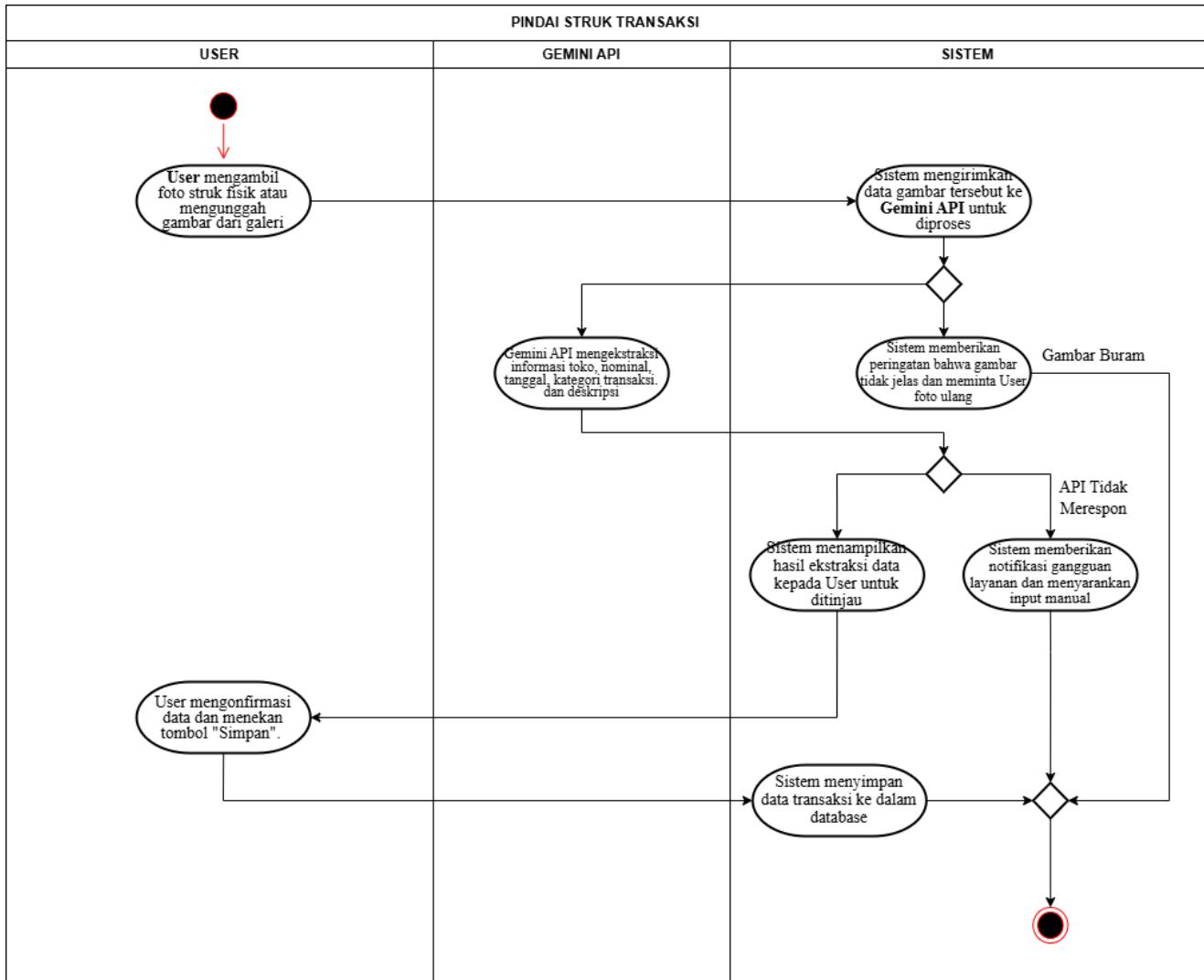
- Mereset KataSandi



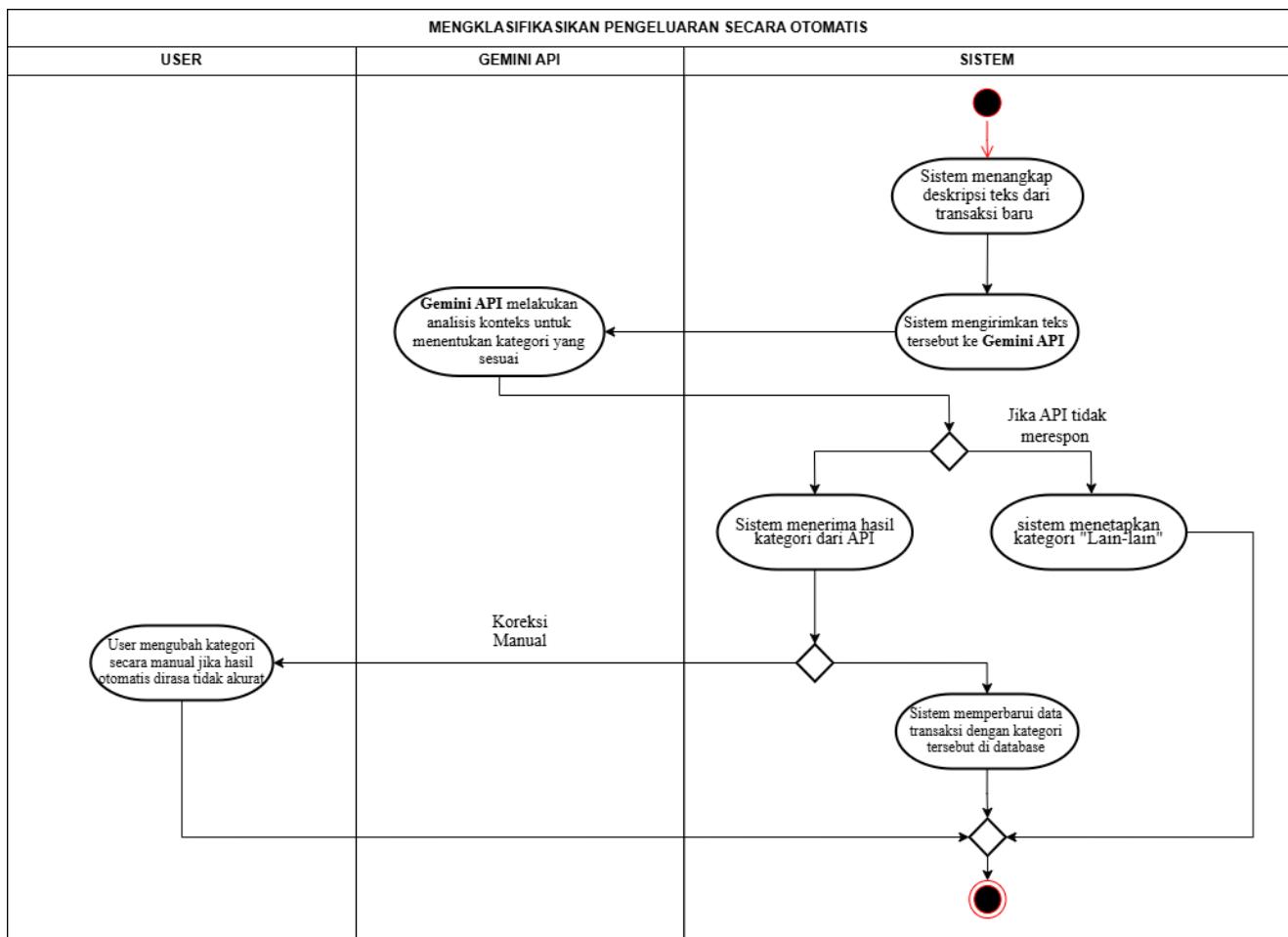
- Mengelola Transaksi



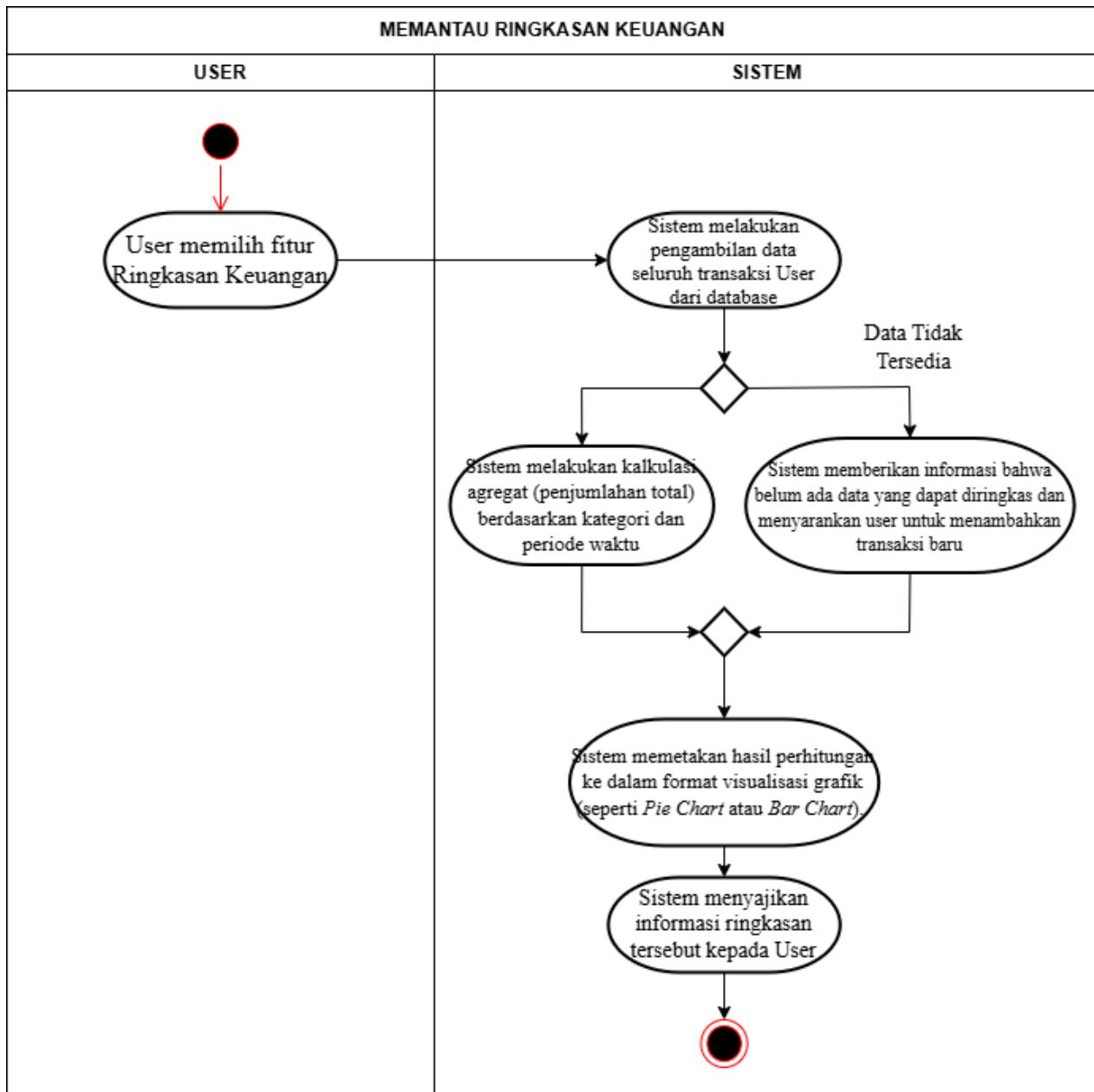
#### ○ Pindai Struk Transaksi



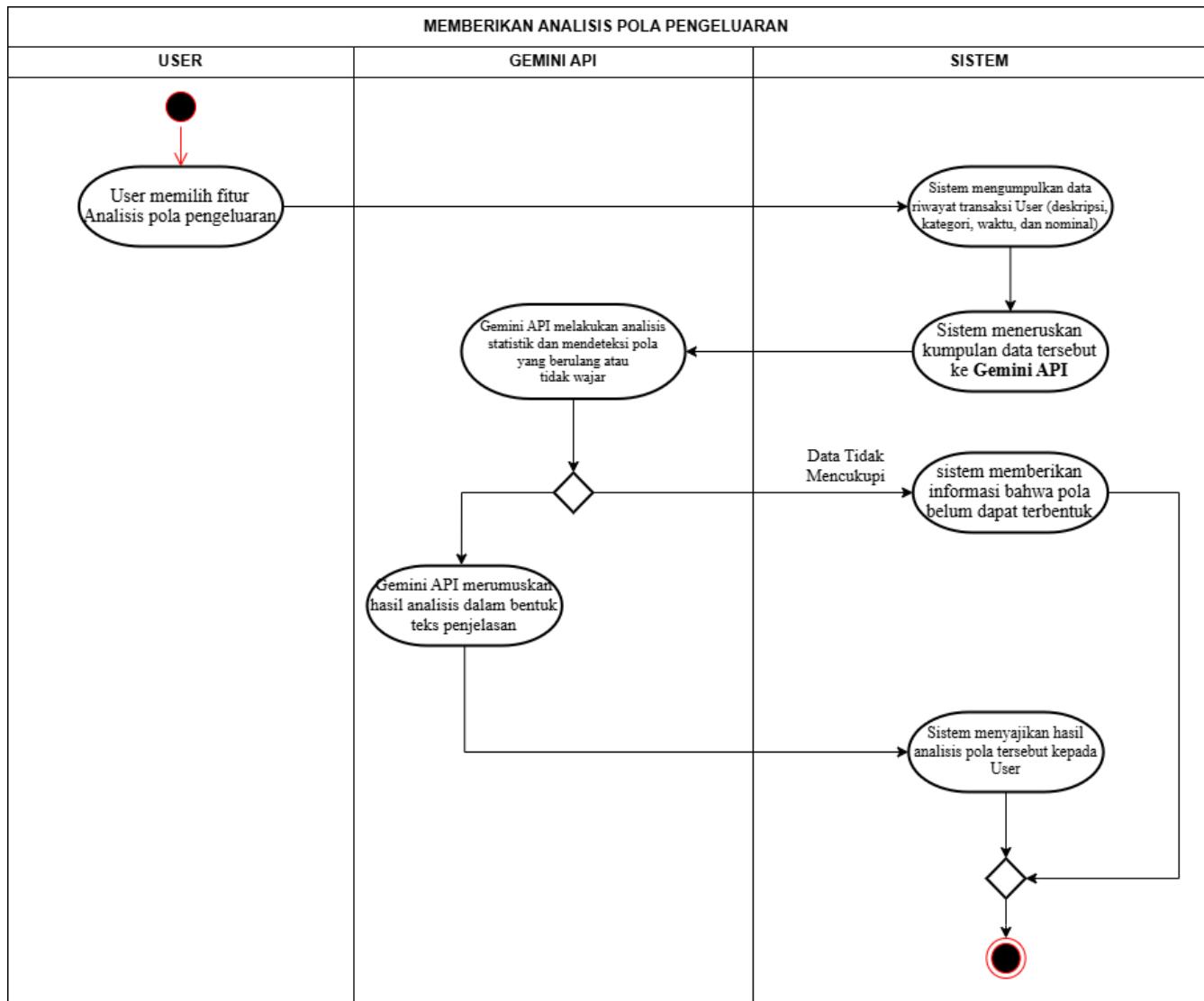
- Mengklasifikasikan Pengeluaran Secara Otomatis



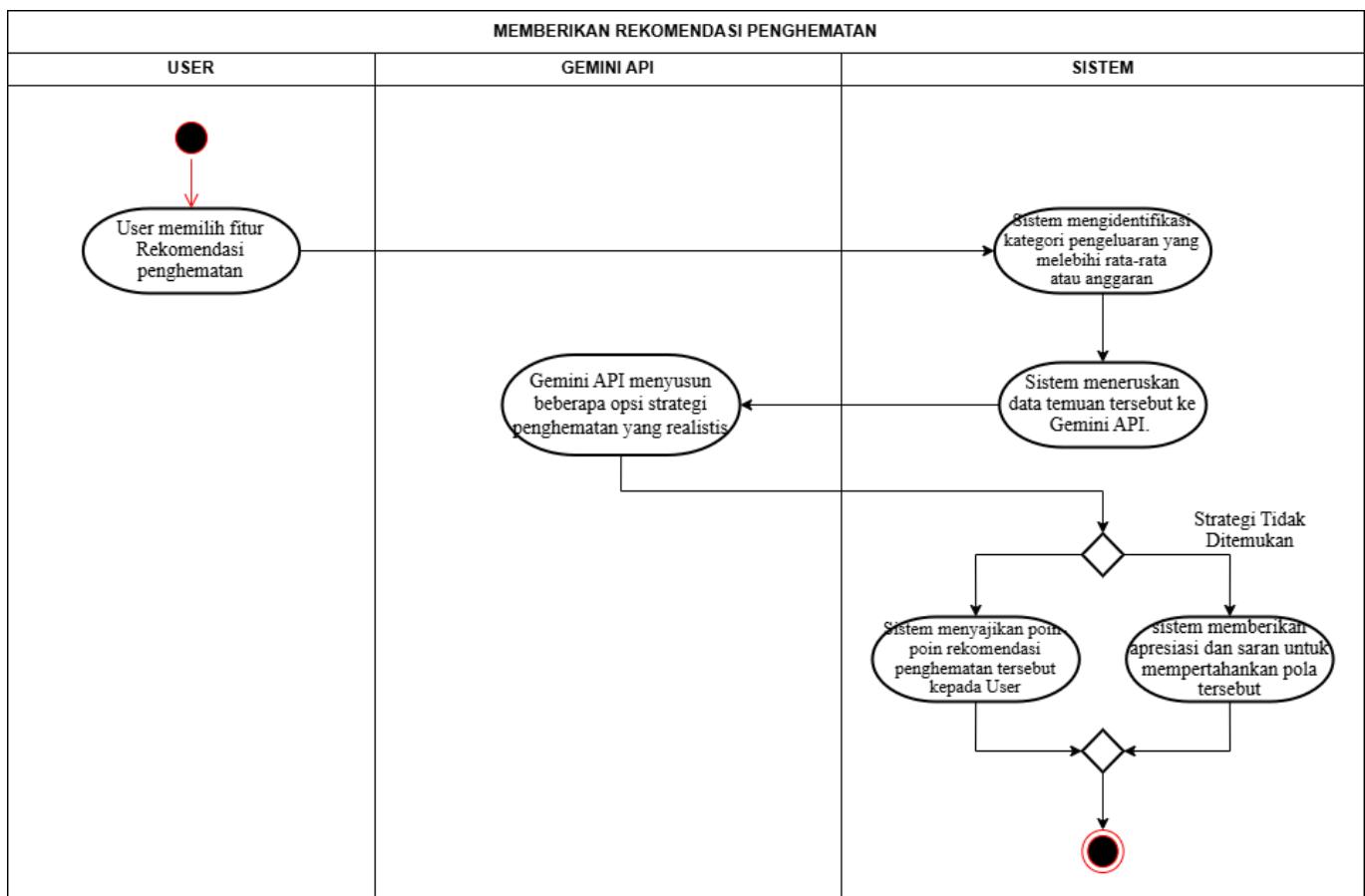
- Memantau Ringkasan Keuangan



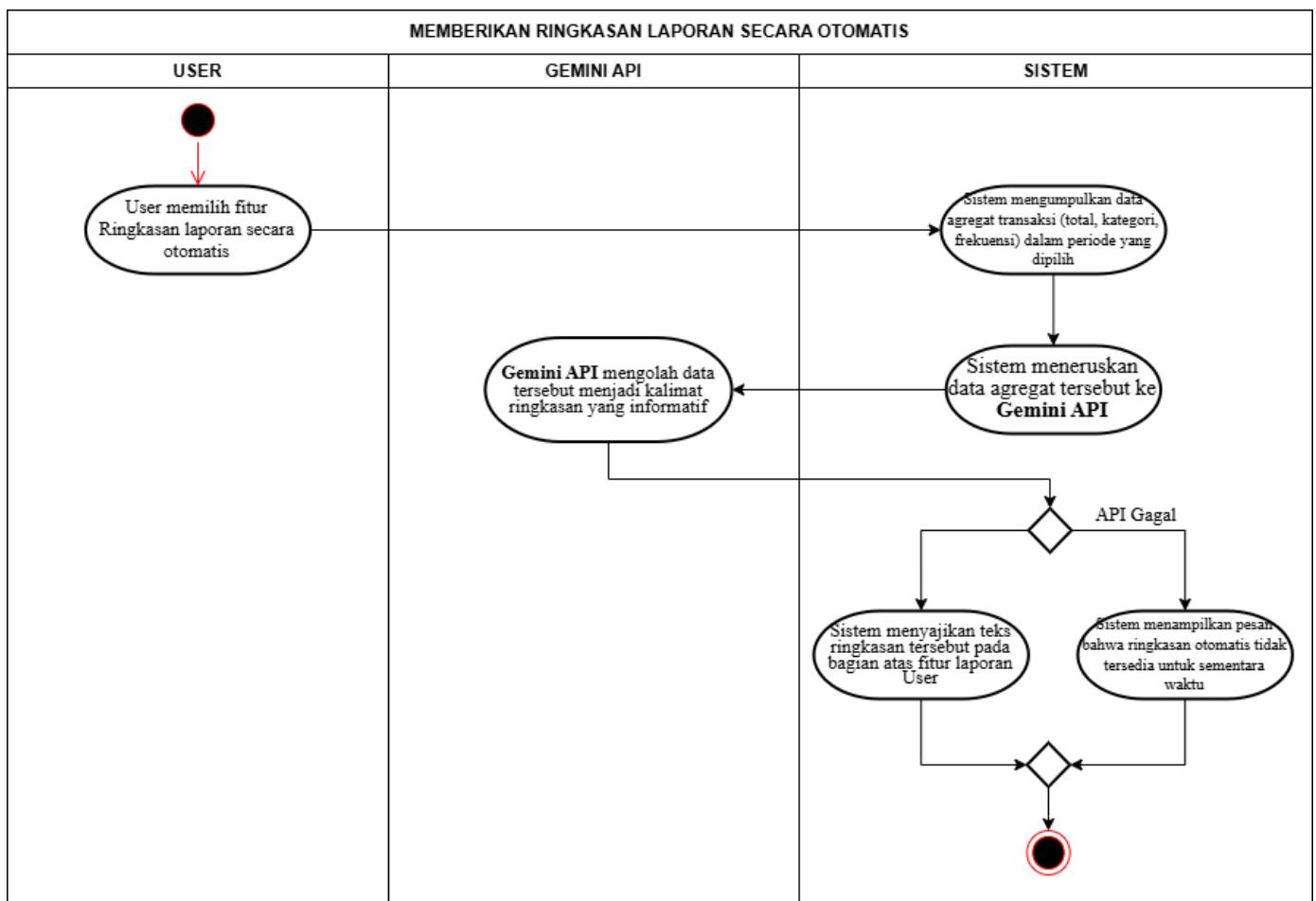
- Memberikan Analisis Pola Pengeluaran



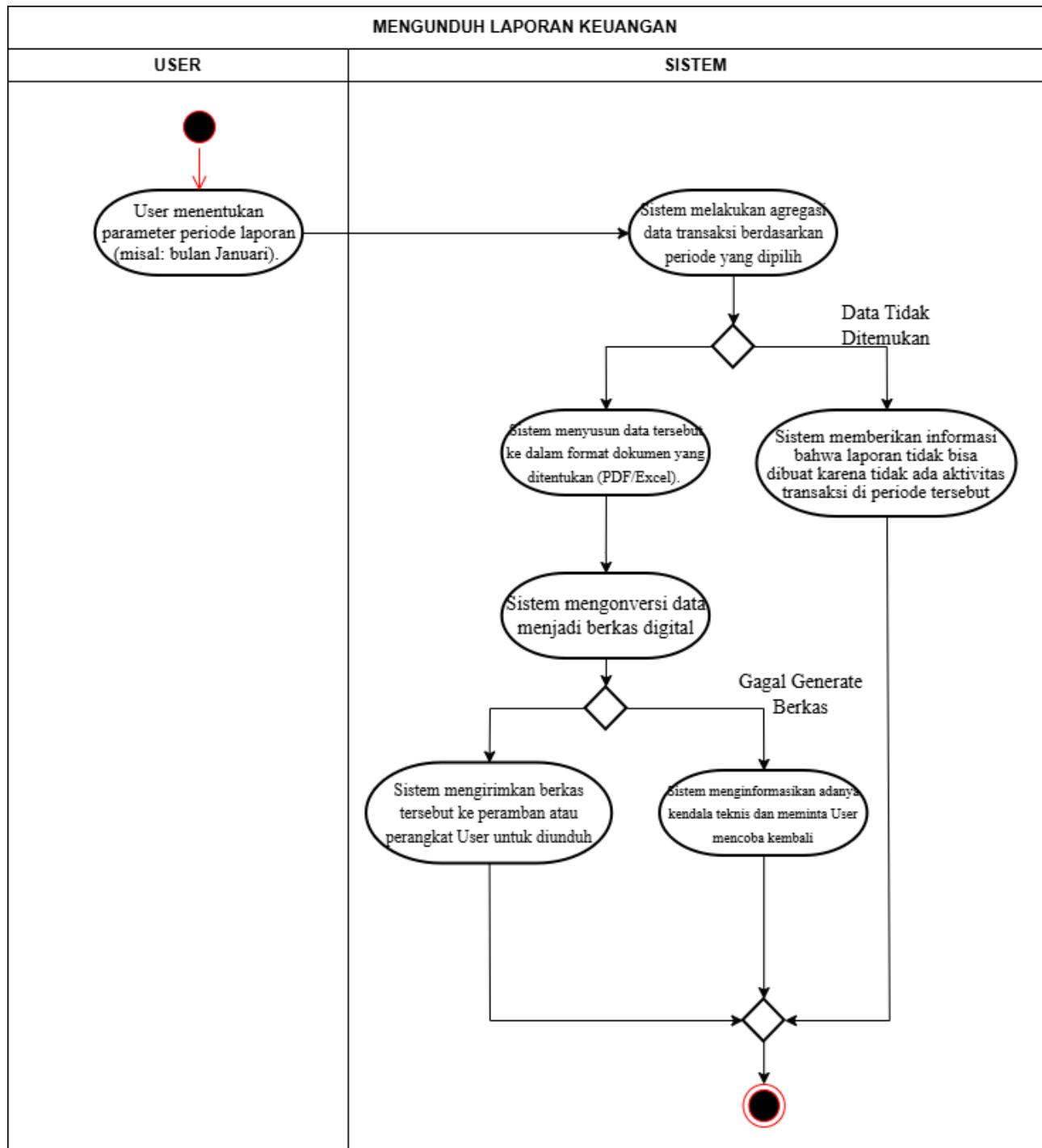
- Memberikan Rekomendasi Penghematan



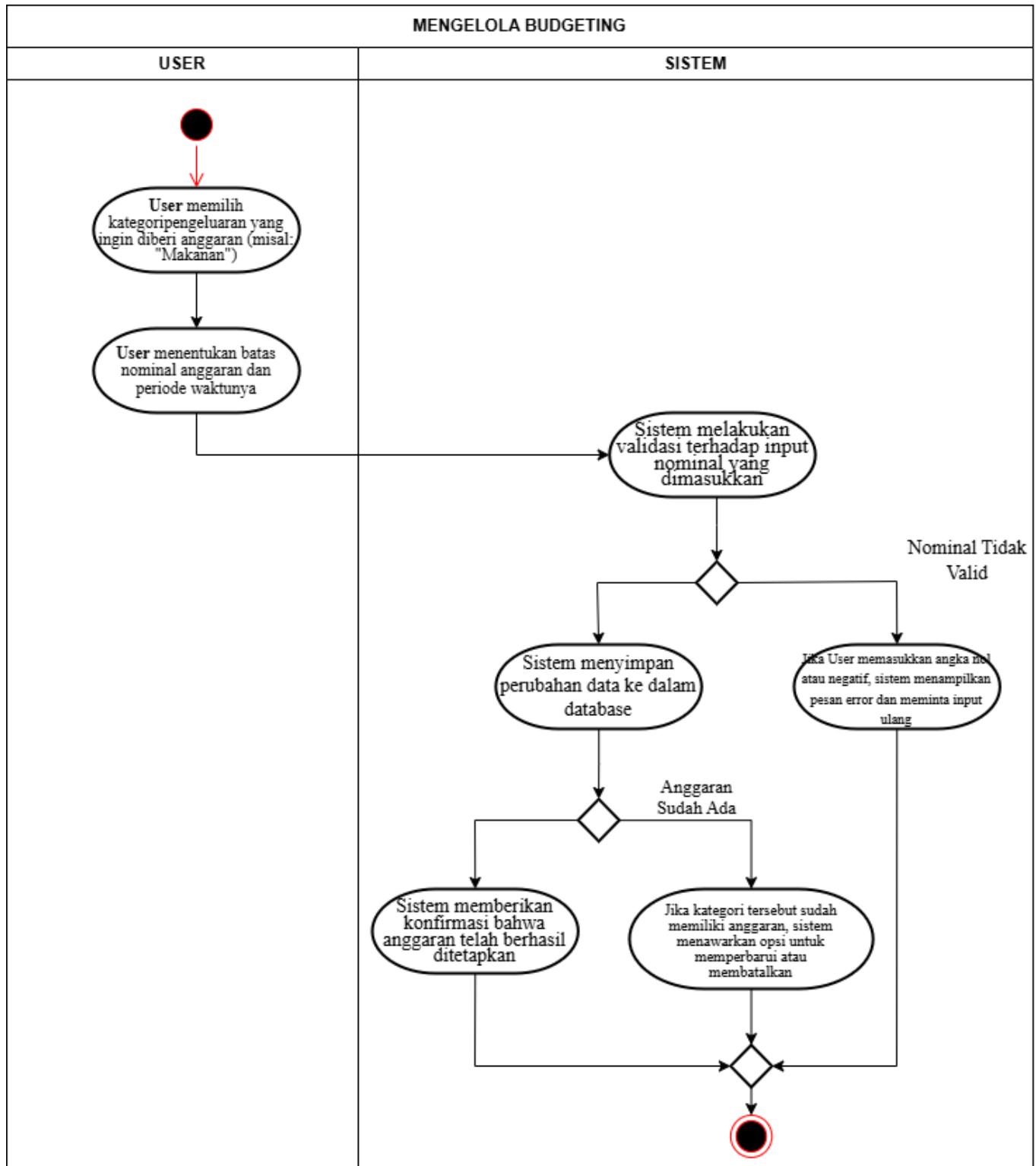
- Memberikan Ringkasan Laporan Secara Otomatis



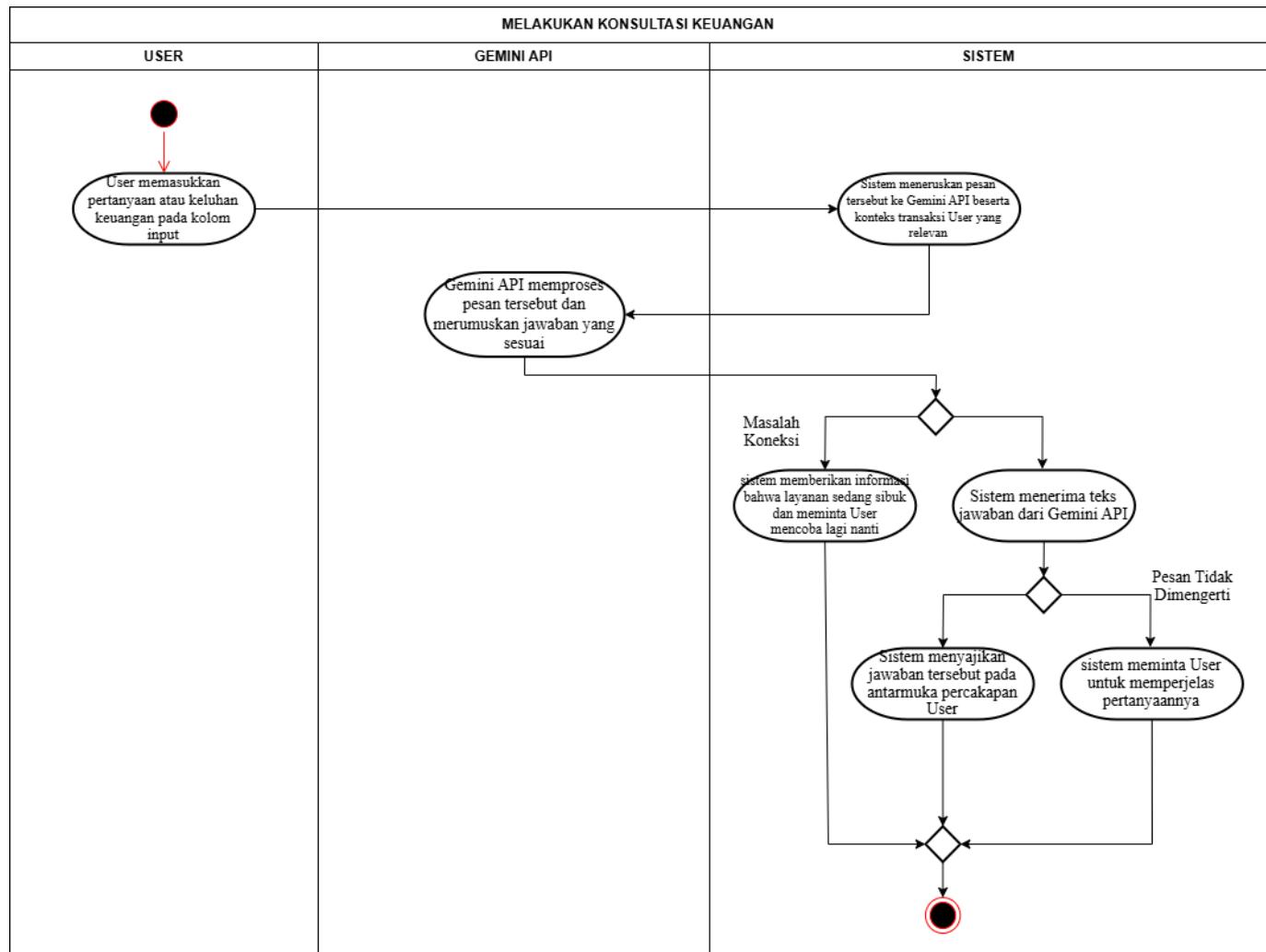
- **Mengunduh Laporan Keuangan**



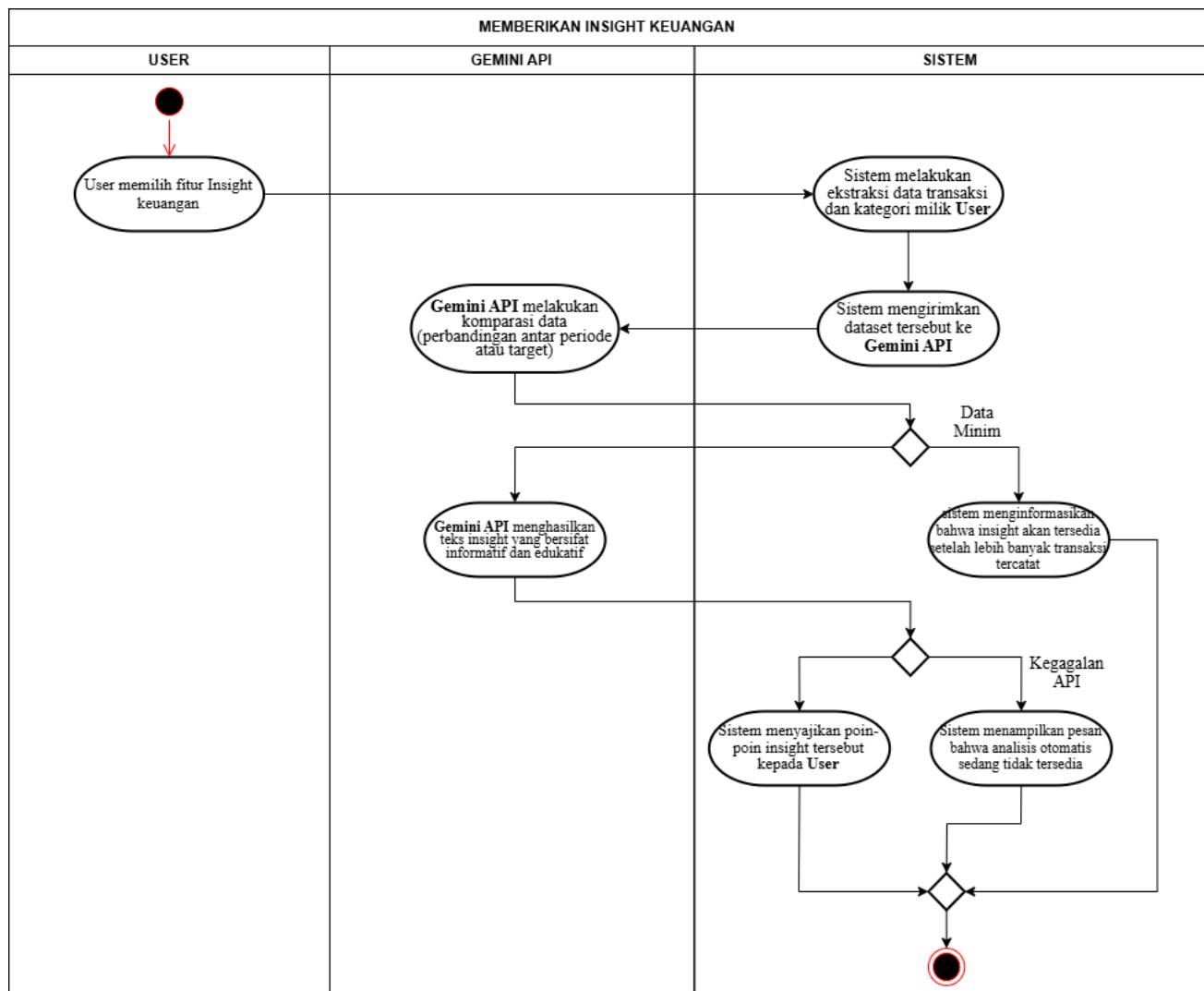
- Mengelola Budgeting



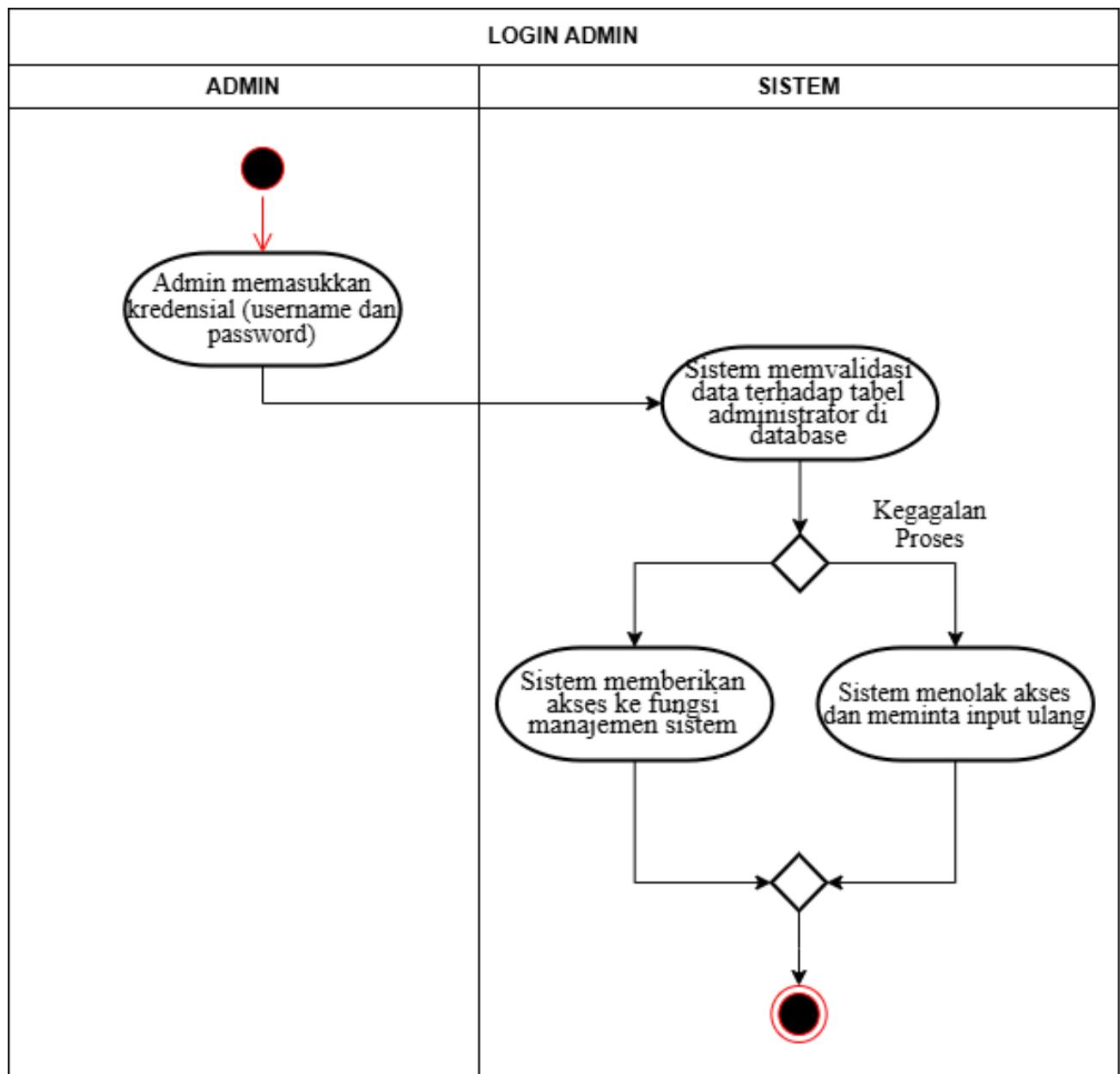
- Melakukan Konsultasi Keuangan



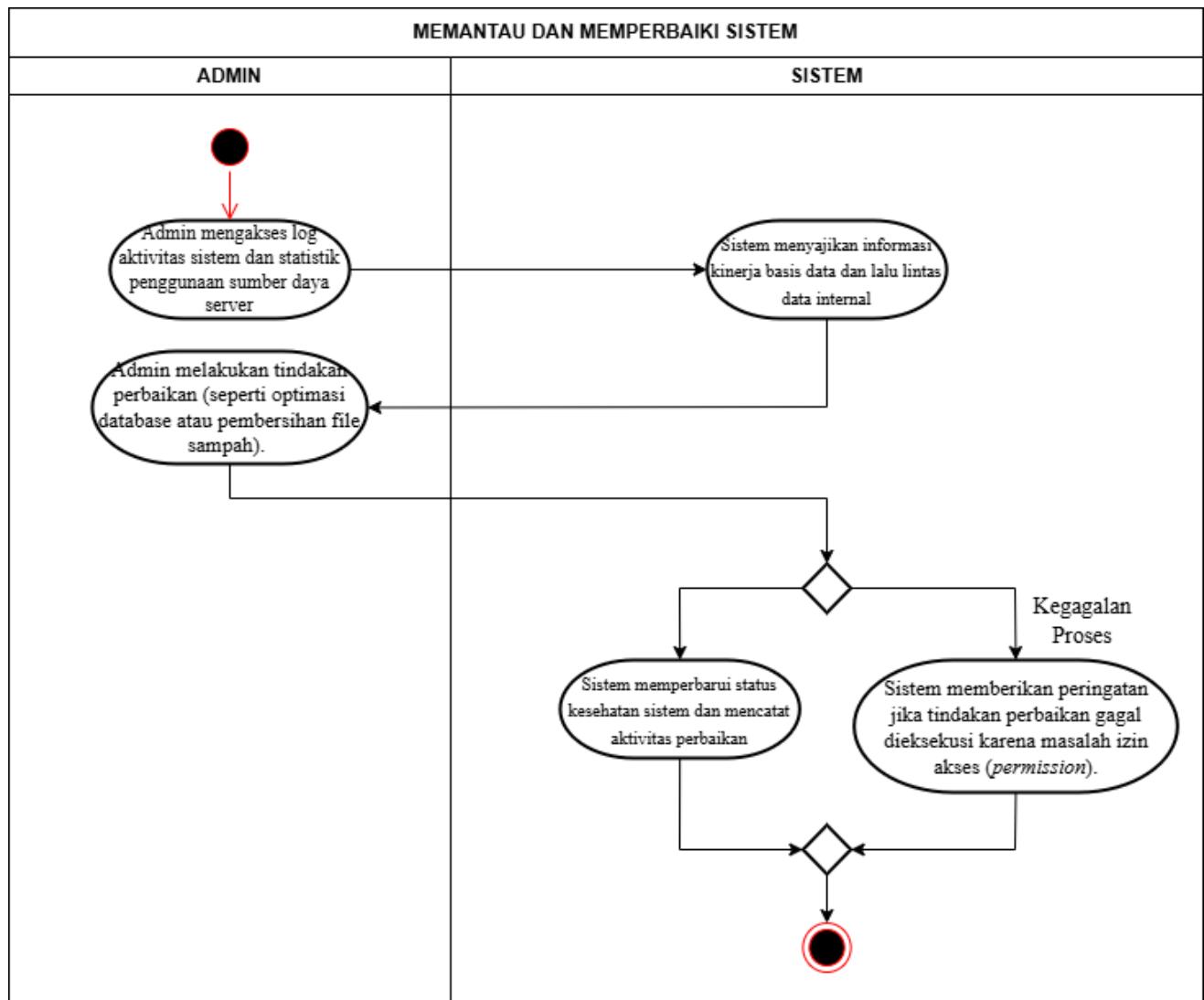
- Memberikan Insight Keuangan



- **Login Admin**



- Memantau dan Memperbaiki Sistem



- Mengelola User

