

# Dokumen Bisnis

*"Roses are red, kerjain TST, or sister-chan won't be your fiancé"*



*H-hah!? W-who wants to be your fiancé anyway, b-baka!? I-it's not like I reminded you to do your TST because I care or anything!... But if you don't finish it, I'll seriously never speak to you again! Jerk!*

***Ringkas.in***

**Ditulis Oleh**

**M. Rayhan Farrukh**

**13523035**

## Daftar Isi

Daftar Isi.....	1
Daftar Gambar.....	1
A. Business Capability Model.....	2
B. Functional Requirement.....	4
C. Use Case.....	6
D. Sequence Diagram.....	11
E. Entity Relationship Diagram.....	13
F. Application Architecture Diagram.....	14
Referensi.....	15

## Daftar Gambar

Gambar 1. Business Capability Model Diagram.....	2
Gambar 2. Use Case Diagram.....	10
Gambar 3. Sequence Diagram UC-01.....	11
Gambar 4. Sequence Diagram UC-02.....	11
Gambar 5. Sequence Diagram UC-03.....	12
Gambar 6. Sequence Diagram UC-04.....	12
Gambar 7. Entity Relationship Diagram.....	13
Gambar 8. Application Architecture Diagram.....	14

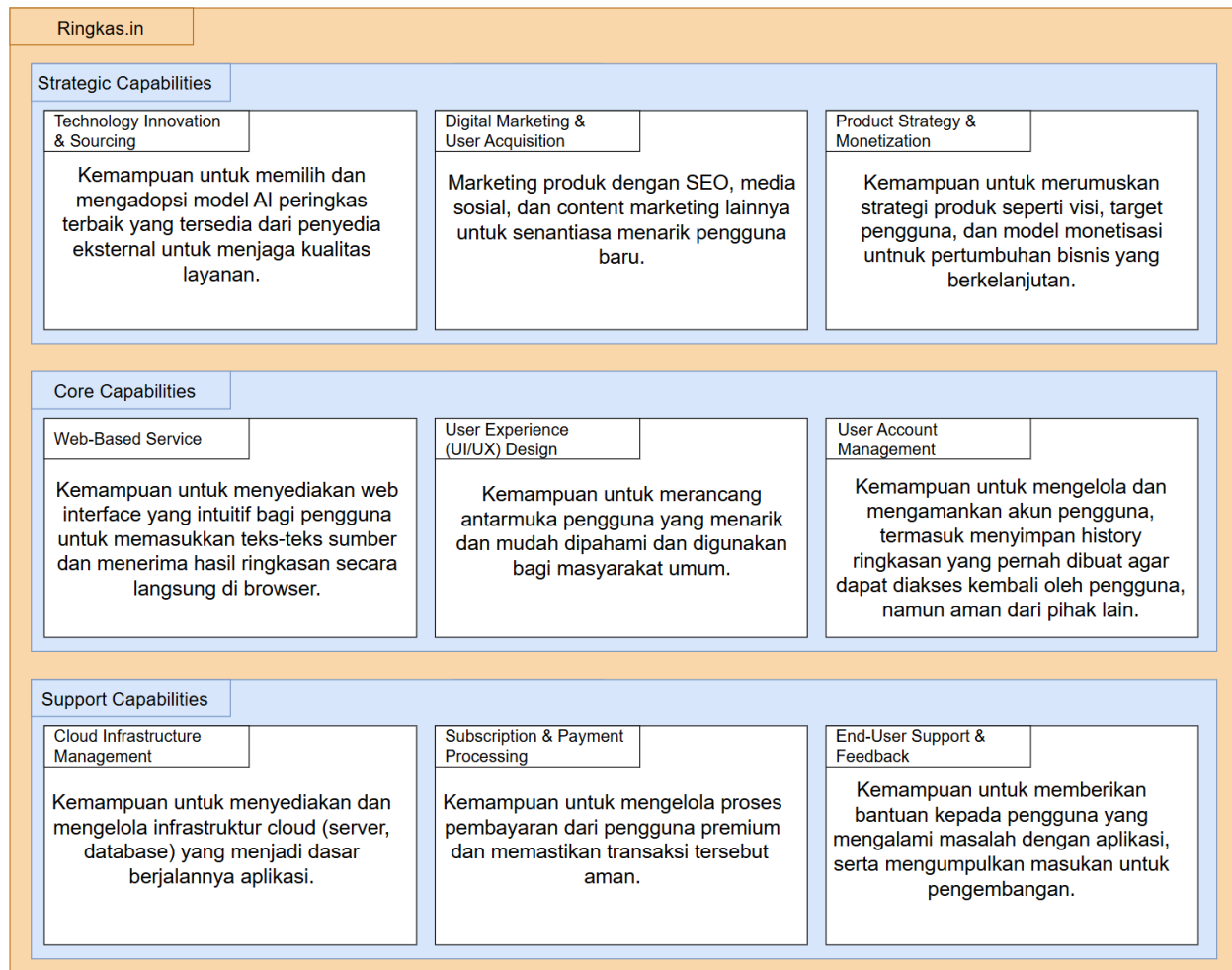
## A. Business Capability Model

Nama Bisnis: Ringkas.in

Deskripsi: Memberikan layanan pembuat ringkasan otomatis, yang dapat menerima banyak teks sumber dari pengguna dan membuat ringkasan dari teks-teks tersebut.

Model Bisnis: Business to Client (B2C)

### Business Capaility Model Diagram



Gambar 1. Business Capability Model Diagram

### Penjelasan Diagram

#### 1. Strategic Capabilities

Kemampuan strategis yang dirancang untuk memenangkan persaingan pasar, serta mempertahankan pertumbuhan jangka panjang.

- *Technology Innovation and Sourcing*, menjaga kualitas layanan agar tetap unggul

sehingga mempertahankan retensi pengguna

- *Digital Marketing and User Acquisition*, senantiasa mendapatkan pengguna baru.
- *Product Strategy and Monetization*, menentukan arah pengembangan dan cara menghasilkan pendapatan.

## *II. Core Capabilities*

Kemampuan utama yang memberikan *value* secara langsung ke pengguna.

- *Web-based Service*, menyediakan fungsi utama layanan peringkasan otomatis yang mudah diakses oleh pengguna.
- *User Experience Design*, membuat antarmuka layanan utama mudah dan enak untuk digunakan oleh semua pengguna.
- *User Account Management*, mengelola identitas dan data pengguna secara aman, agar integritas data pengguna aman.

## *III. Support Capabilities*

Kemampuan internal perusahaan untuk membantu memastikan kelancaran operasional layanan pada aplikasi.

- *Cloud Infrastructure Management*, memastikan infrastruktur teknis berjalan dengan andal
- *Subscription and Payment Processing*, mengelola proses transaksi secara aman, efisien, dan simpel bagi pengguna.
- *End-User Support and Feedback*, memberikan bantuan kepada pengguna serta mengumpulkan *feedback* untuk pengembangan produk sesuai yang diinginkan pengguna.

## **B. Functional Requirement**

### **FR-01 Pengelolaan Sesi Pengguna**

#### **Deskripsi**

Sistem harus memungkinkan pengguna untuk mengelola sesi aplikasi mereka secara aman, yang mencakup proses masuk (login) menggunakan akun eksternal yang sudah ada dan proses keluar (logout).

#### **Kriteria Keberhasilan**

- Pengguna dapat memulai proses login dari halaman utama.
- Login harus menggunakan protokol OAuth 2.0 dengan penyedia layanan eksternal (misalnya Google).
- Setelah login berhasil, sistem membuat sesi aktif yang terasosiasi dengan identitas pengguna.
- Sistem memiliki fungsi logout untuk mengakhiri sesi aktif pengguna.
- Fungsionalitas inti akan dibatasi untuk pengguna yang tidak memiliki sesi aktif.

### **FR-02 Pembuatan Ringkasan Otomatis**

#### **Deskripsi**

Sistem harus mampu menerima masukan teks dari pengguna yang terautentikasi, memprosesnya untuk menghasilkan ringkasan, dan menampilkan hasilnya kepada pengguna.

#### **Kriteria Keberhasilan:**

- Sistem menyediakan area input, atau tombol *upload* dokumen di antarmuka untuk pengguna memasukkan teks.
- Setelah dimulai, sistem harus mengolah dan mengirimkan teks masukan ke API LLM eksternal untuk diproses menjadi ringkasan yang baik.
- Sistem harus menampilkan hasil ringkasan yang diterima dari API LLM di antarmuka pengguna.
- Sistem secara otomatis menyimpan teks asli dan hasil ringkasan ke riwayat ringkasan milik pengguna.

### **FR-03 Peninjauan Riwayat Ringkasan**

#### **Deskripsi**

Sistem harus memungkinkan pengguna yang terautentikasi untuk melihat daftar semua ringkasan yang pernah mereka buat sebelumnya.

#### Kriteria Keberhasilan

- Sistem harus menyediakan halaman atau bagian "Riwayat" yang hanya dapat diakses oleh pengguna yang sudah login.
- Pada halaman tersebut, sistem harus menampilkan daftar ringkasan yang hanya dimiliki oleh pengguna yang sedang aktif.
- Pengguna harus dapat memilih salah satu entri riwayat untuk melihat kembali teks asli dan hasil ringkasan secara lengkap.
- Pengguna harus dapat menghapus sebuah entri pada riwayat mereka.

## C. Use Case

### UC-01 Mengelola Sesi Pengguna

Aktor: Pengguna

Kode Functional Requirements: FR-01

Deskripsi: *Use case* ini mencakup alur bagi pengguna untuk masuk dan keluar dari akun pada aplikasi.

#### Skenario Use Case

- Skenario 1: Pengguna berhasil melakukan *login*  
Kondisi Awal: Pengguna berada di halaman utama dan belum *login*  
Kondisi Akhir: Pengguna berada di halaman membuat ringkasan dengan sesi aktif

Aksi Aktor	Reaksi Sistem
1. Pengguna menekan tombol "Login"	
	2. Sistem mengarahkan pengguna ke halaman autentikasi eksternal
3. Pengguna memasukkan kredensial yang <i>valid</i>	
	4. Sistem menerima konfirmasi dan membuat sesi aktif untuk pengguna

- Skenario 2: Pengguna gagal melakukan *login*  
Kondisi Awal: Pengguna berada di halaman autentikasi  
Kondisi Akhir: Pengguna tetap pada halaman autentikasi tanpa sesi aktif

Aksi Aktor	Reaksi Sistem
1. Pengguna memasukkan kredensial yang tidak <i>valid</i>	
	2. Sistem mendeteksi kesalahan kredensial
	3. Sistem menampilkan pesan tentang <i>invalid credentials</i>

- Skenario 3: Pengguna melakukan *logout*

Kondisi Awal: Pengguna sudah *login* dan memiliki sesi aktif

Kondisi Akhir: Pengguna kembali ke halaman utama tanpa sesi aktif

Aksi Aktor	Reaksi Sistem
1. Pengguna menekan tombol "Logout"	
	2. Sistem menghapus sesi aktif pengguna dan mengembalikannya ke halaman utama

## UC-02 Membuat Ringkasan

Aktor: Pengguna

Kode Functional Requirements: FR-02

Deskripsi: *Use case* ini mencakup alur fungsionalitas utama aplikasi, yaitu pembuatan ringkasan dari sumber teks yang dimasukkan pengguna.

### Skenario Use Case

- Skenario 1: Berhasil Membuat Ringkasan

Kondisi Awal: Pengguna berada di halaman utama peringkasan.

Kondisi Akhir: Ringkasan baru berhasil dibuat dan tersimpan di riwayat pengguna

Aksi Aktor	Reaksi Sistem
1. Pengguna memasukkan teks ke dalam area input yang disediakan.	
2. Pengguna menekan tombol untuk memulai ringkasan	
	3. Sistem mengirim teks ke API eksternal
	4. Sistem menerima hasil dari API eksternal, menampilkannya kepada pengguna, serta menyimpan ke dalam riwayat

- Skenario 2: Gagal Membuat Ringkasan

Kondisi Awal: Pengguna berada di halaman utama peringkasan.

Kondisi Akhir: Sistem menampilkan pesan *error*



Aksi Aktor	Reaksi Sistem
1. Pengguna memasukkan teks ke dalam area input yang disediakan.	
2. Pengguna menekan tombol untuk memulai ringkasan	
	3. Sistem mengirim teks ke API eksternal
	4. Sistem mendapat pesan <i>error</i> dari API eksternal, atau tidak mendapat respons sama sekali
	5. Sistem menampilkan pesan <i>error</i> kepada pengguna

### UC-03 Melihat Riwayat Ringkasan

Aktor: Pengguna

Kode Functional Requirements: FR-03

Deskripsi: *Use case* ini mencakup alur saat pengguna ingin melihat ringkasan yang pernah dibuat sebelumnya.

#### Skenario Use Case

- Skenario 1: Pengguna memiliki riwayat  
Kondisi Awal: Pengguna memiliki sesi aktif  
Kondisi Akhir: Sistem menampilkan riwayat ringkasan pengguna

Aksi Aktor	Reaksi Sistem
1. Pengguna menavigasi ke halaman riwayat	
	2. Sistem mengambil dan menampilkan daftar riwayat ringkasan pengguna
3. Pengguna memilih salah satu entri dari daftar	
	4. Sistem menampilkan teks asli dan hasil ringkasan dari entri yang dipilih

- Skenario 2: Pengguna belum memiliki riwayat  
Kondisi Awal: Pengguna memiliki sesi aktif  
Kondisi Akhir: Sistem menampilkan pesan bahwa pengguna belum memiliki riwayat

Aksi Aktor	Reaksi Sistem
1. Pengguna menavigasi ke halaman riwayat	
	2. Sistem menampilkan pesan "Anda belum memiliki riwayat."

#### UC-04 Menghapus Entri Riwayat Ringkasan

Aktor: Pengguna

Kode Functional Requirements: FR-03

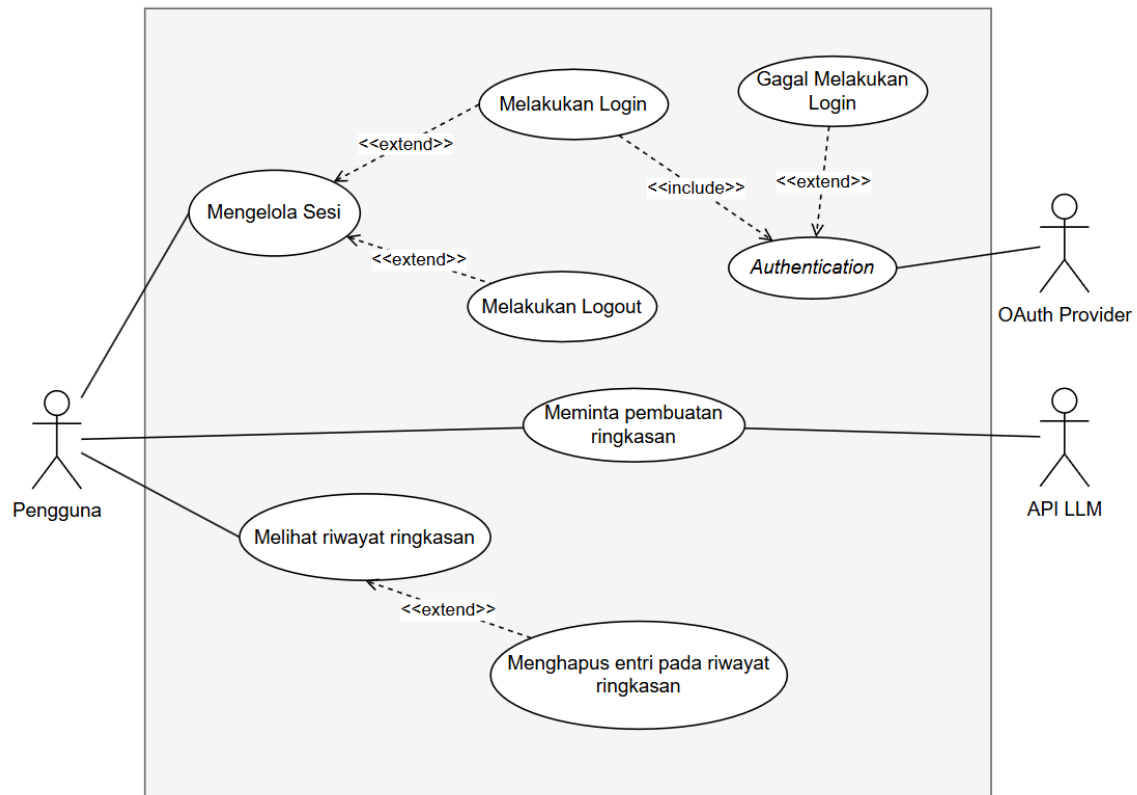
Deskripsi: *Use case* ini mencakup alur ketika pengguna ingin menghapus entri pada riwayat ringkasan.

##### Skenario Use Case

- Skenario 1: Berhasil menghapus entri  
Kondisi Awal: Pengguna memiliki sesi aktif dan berada di halaman riwayat  
Kondisi Akhir: Sistem menghapus entri riwayat yang diinginkan pengguna

Aksi Aktor	Reaksi Sistem
1. Pengguna menekan tombol "Hapus" pada salah satu entri riwayat	
	2. Sistem menampilkan dialog konfirmasi untuk penghapusan
3. Pengguna mengkonfirmasi penghapusan.	
	4. Sistem menghapus entri dari database, dan memperbarui tampilan daftar riwayat.

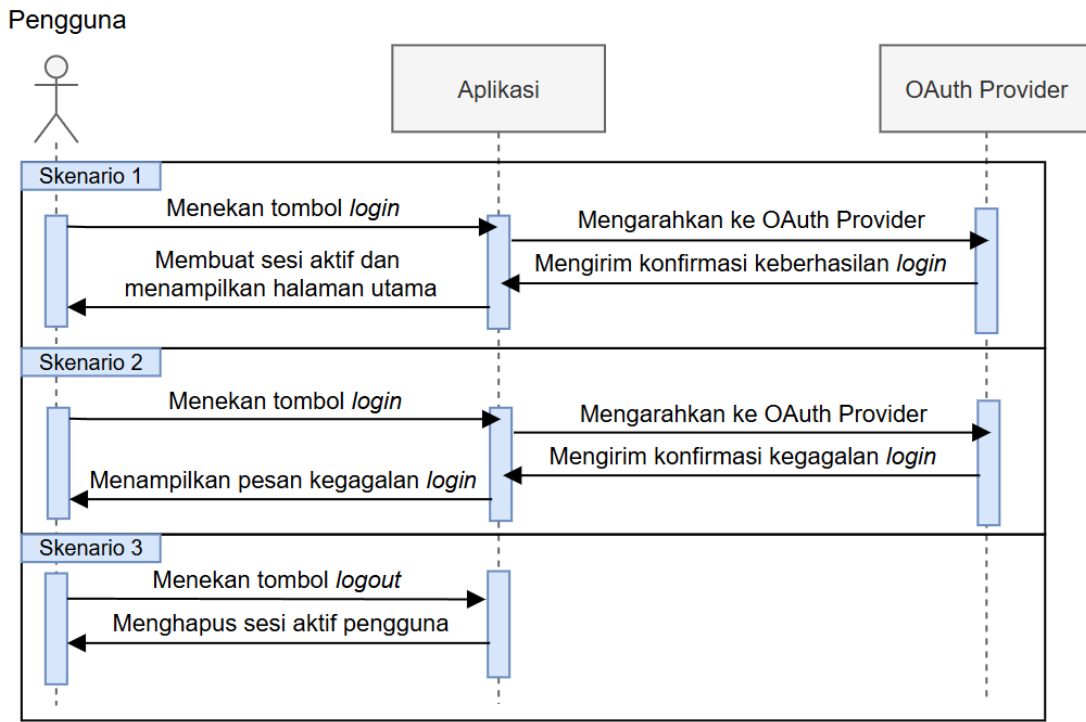
## Diagram Use Case



Gambar 2. Use Case Diagram

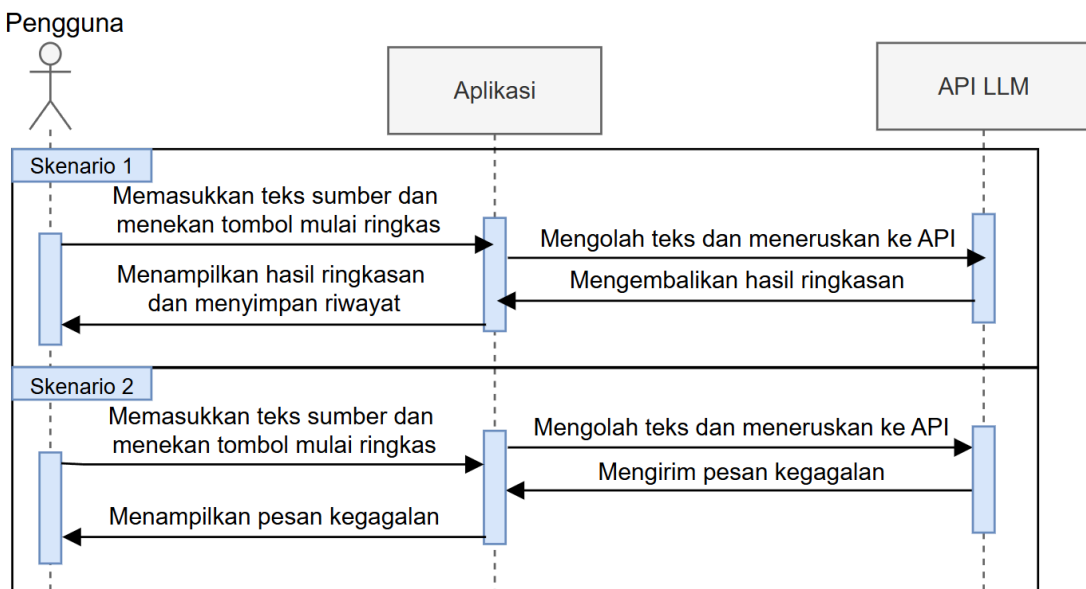
## D. Sequence Diagram

### Sequence Diagram UC-01



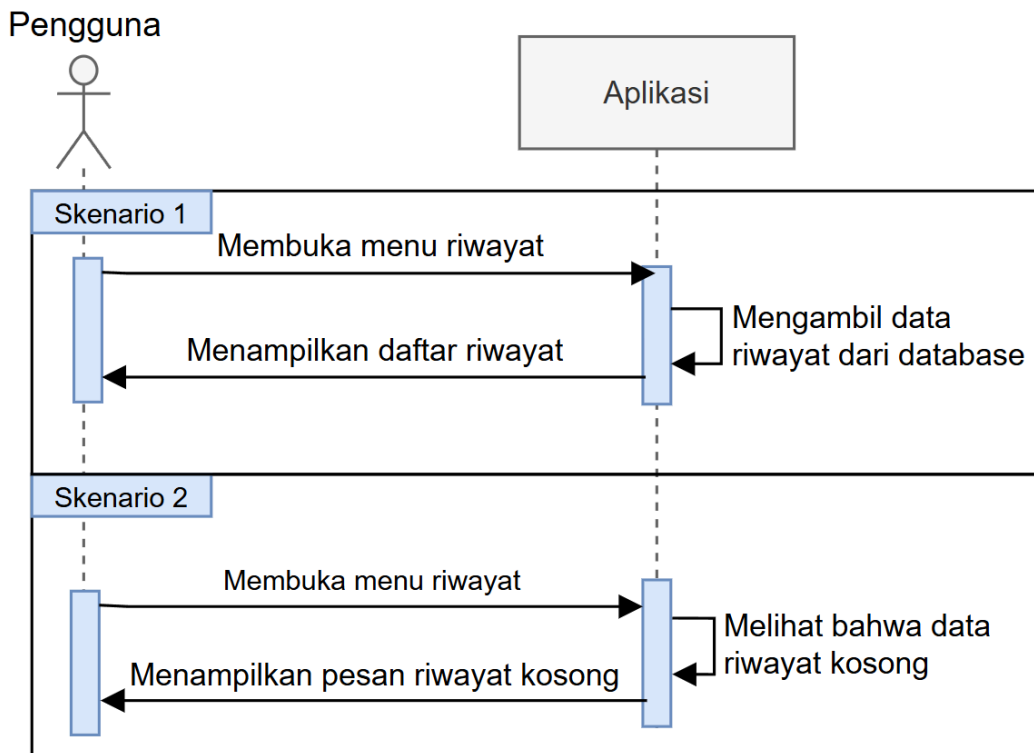
Gambar 3. Sequence Diagram UC-01

### Sequence Diagram UC-02



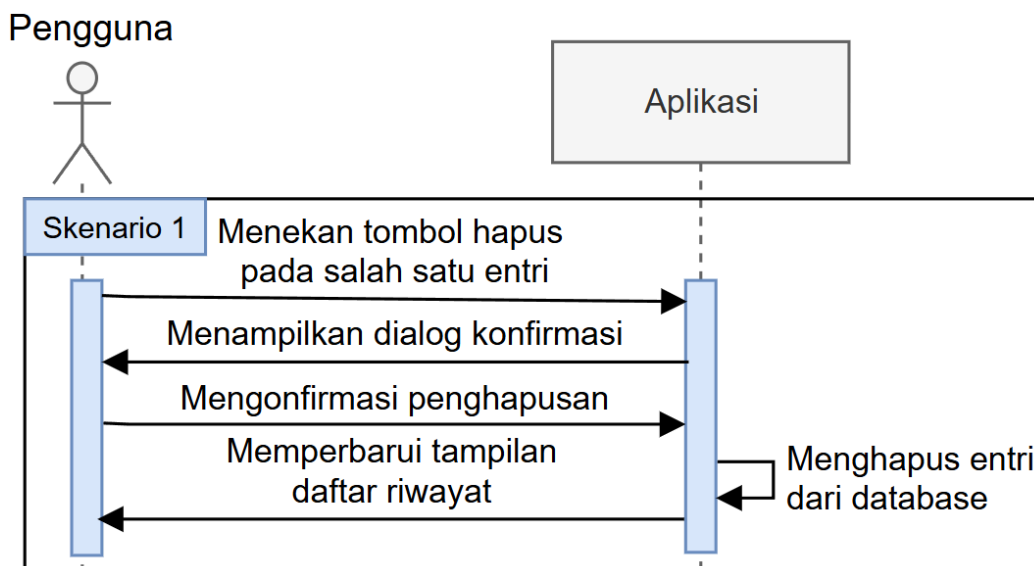
Gambar 4. Sequence Diagram UC-02

### Sequence Diagram UC-03



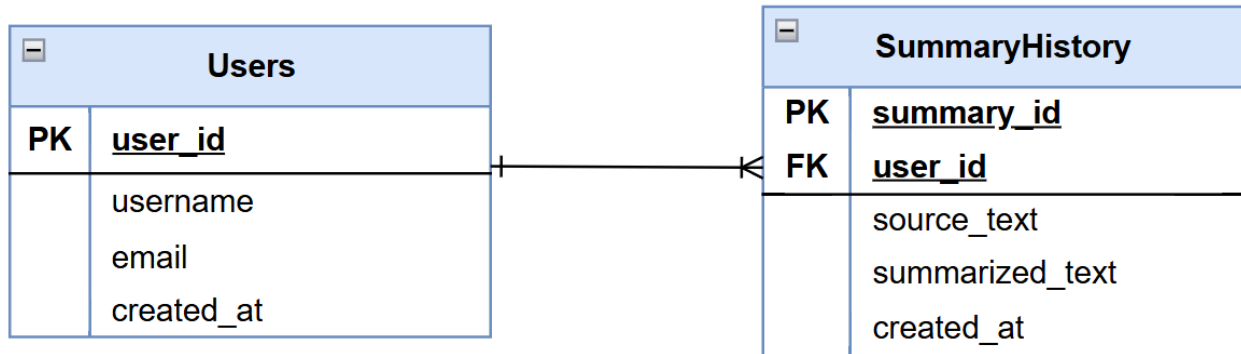
Gambar 5. Sequence Diagram UC-03

### Sequence Diagram UC-04



Gambar 6. Sequence Diagram UC-04

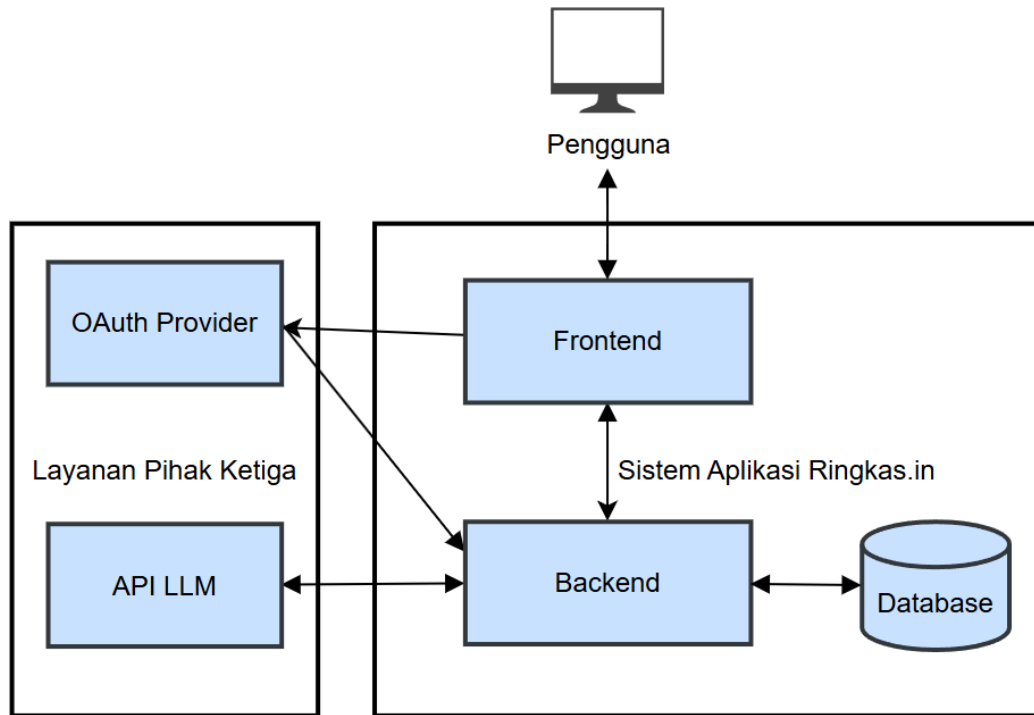
## E. Entity Relationship Diagram



Gambar 7. Entity Relationship Diagram

*Entity Relationship Diagram* diatas menggambarkan arsitektur data aplikasi. Data yang perlu disimpan aplikasi hanyalah data pengguna (*Users*) dan riwayat ringkasannya (*SummaryHistory*). Kedua *entity* terhubung dengan relasi *one-to-many* dimana satu pengguna bisa memiliki banyak entri riwayat, namun satu entri riwayat unik hanya dimiliki oleh tepat satu pengguna.

## F. Application Architecture Diagram



Gambar 8. Application Architecture Diagram

Diagram diatas menggambarkan arsitektur dari aplikasi [Ringkas.in](https://ringkas.in), yang menunjukkan hubungan komponen-komponen di dalam sistem, beserta *flow* datanya. Arah panah menunjukkan *data flow*, sedangkan kotak dan tabung berwarna menunjukkan komponen sistem.

Di luar sistem terdapat pengguna yang berinteraksi dengan sistem melalui antarmuka *frontend*. *Frontend* aplikasi lah yang kemudian terhubung dengan *backend* yang akan memproses aksi-aksi yang dilakukan pengguna. Selain itu, *frontend* juga terhubung ke *OAuth Provider* (misalnya ketika *frontend* mengarahkan pengguna ke halaman *login* google). Di sisi lain, *backend* aplikasilah yang memproses aksi-aksi pengguna dan meneruskan *input* ke API eksternal atau memperbarui *database* pada sistem.

## Referensi

- [1] "Understanding and Creating an ArchiMate Capability Map in TOGAF," Visual Paradigm, n.d. [Online]. Available: <https://guides.visual-paradigm.com/understanding-and-creating-an-archimate-capability-map-in-togaf>
- [2] "UML Use Case Diagram Tutorial," GeeksforGeeks, n.d. [Online]. Available: <https://www.geeksforgeeks.org/system-design/use-case-diagram>
- [3] "UML Sequence Diagrams in Software Engineering," GeeksforGeeks, n.d. [Online]. Available: <https://www.geeksforgeeks.org/unified-modeling-language-uml-sequence-diagrams>
- [4] "Creating Sequence Diagrams with a Use Case Driven Approach: A Comprehensive Guide," Visual Paradigm, n.d. [Online]. Available: <https://guides.visual-paradigm.com/creating-sequence-diagrams-with-a-use-case-driven-approach-a-comprehensive-guide>
- [5] M. M., "ER Diagram," Lucidchart, n.d. [Online]. Available: <https://www.lucidchart.com/pages/er-diagrams>
- [6] Amazon Web Services, Inc., "What is architecture diagramming?," AWS, n.d. [Online]. Available: <https://aws.amazon.com/what-is/architecture-diagramming>