

Halaman Cover

SOFTWARE REQUIREMENT SPECIFICATION

Berdasarkan IEEE Concept of Operations

Untuk Mata Kuliah Desain dan Analisis Sistem

Dosen Pengampu:

Renny Sari Dewi, S.Kom., M.Kom., MCE, MOS

Program Studi Bisnis Digital

Fakultas Ekonomika dan Bisnis

Universitas Negeri Surabaya

2025

Pembagian Tugas dan Tanggungjawab

NIM	Nama Lengkap	Rincian Tugas dan Tanggungjawab

1. High-level Requirements dan Domain Object

1.1.High-level Requirements

High-Level Requirements (HLR) – Sistem Informasi Pelayanan Katering

Sistem catering akan berbasis web pada awalnya, tetapi harus memiliki arsitektur yang fleksibel sehingga alternatif antarmuka pengguna dapat dikembangkan (aplikasi mobile, layanan web, dll.).

1. Sistem katering akan berbasis web, tetapi harus memiliki arsitektur yang fleksibel, sehingga alternatif antarmuka pengguna dapat dikembangkan, seperti aplikasi mobile atau layanan web lainnya.
2. Sistem harus dapat menerima pesanan makanan, dengan order yang dilakukan secara real-time melalui internet.
3. Pengguna harus dapat melihat daftar menu, termasuk deskripsi, harga, komposisi bahan, dan foto makanan.
4. Pengguna harus dapat menambahkan makanan ke dalam keranjang belanja sebelum melakukan checkout.
5. Pengguna harus dapat menghapus item dari keranjang belanja sebelum melakukan konfirmasi pemesanan.
6. Pengguna harus dapat memilih jadwal pengiriman sesuai dengan ketersediaan layanan.
7. Sistem harus dapat mengelola jadwal pemesanan berulang bagi pelanggan yang ingin berlangganan katering secara harian, mingguan, atau bulanan.
8. Pengguna harus dapat melakukan pembatalan pesanan sebelum pesanan tersebut diproses atau dikirim.
9. Sistem harus mendukung berbagai metode pembayaran, termasuk kartu kredit, transfer bank, dan dompet digital.
10. Sistem harus menyediakan fitur konfirmasi pembayaran secara otomatis atau manual oleh admin.
11. Sistem harus memberikan notifikasi kepada pengguna melalui email atau SMS, terkait status pesanan, pembayaran, dan pengiriman.
12. Sistem harus memungkinkan pelanggan untuk melacak status pesanan, mulai dari pemrosesan, persiapan, hingga pengiriman.

13. Sistem harus menyediakan **dashboard admin** untuk mengelola pesanan, pelanggan, dan **laporan penjualan**.
14. Sistem harus memungkinkan admin untuk menambah, mengedit, atau menghapus **menu catering**.
15. Sistem harus memungkinkan admin untuk mengelola **stok bahan makanan**, sehingga ketersediaan bahan baku dapat dipantau.
16. Sistem harus menyediakan **laporan transaksi** dan grafik penjualan bagi admin dan pemilik bisnis, dalam format harian, mingguan, atau bulanan.
17. Sistem harus memungkinkan admin untuk mengelola **diskon** atau **promo**, yang dapat diterapkan pada pesanan tertentu.
18. Sistem harus memiliki fitur **voucher** atau **kupon**, yang dapat digunakan pelanggan untuk mendapatkan potongan harga.
19. Sistem harus dapat diintegrasikan ke dalam **website mitra** melalui **mini-katalog**, yang berasal dari katalog utama dalam **database pusat**.
20. Sistem harus mendukung fitur **live chat** atau **chatbot**, untuk membantu pelanggan dalam pemrosesan pemesanan dan pertanyaan lainnya.
21. Sistem harus memiliki mekanisme **cadangan data otomatis**, untuk mencegah kehilangan data penting.
22. Sistem harus menerapkan **keamanan data pelanggan**, termasuk enkripsi **informasi pembayaran** dan kebijakan privasi yang jelas.
23. Sistem harus memiliki **kontrol akses multi-level**, dengan perbedaan hak akses antara admin, staff, dan pelanggan.
24. Sistem harus menyediakan fitur **penjadwalan catering** untuk acara khusus, seperti pesta, pernikahan, atau rapat.
25. Sistem harus dapat memberikan **rekomendasi menu** kepada pelanggan berdasarkan **riwayat pemesanan** mereka.
26. Pengguna harus dapat melihat **riwayat transaksi** dan pesanan sebelumnya dalam akun mereka.
27. Pengguna harus dapat memberikan **ulasan** dan **rating** terhadap makanan dan layanan yang diterima.
28. Pengguna harus dapat menyimpan lebih dari satu **alamat pengiriman**, untuk mempermudah pemesanan ke lokasi yang berbeda.
29. Pengguna harus dapat menghubungi admin atau **customer service** melalui fitur live chat, untuk mendapatkan bantuan dalam pemesanan.

30. Pengguna harus dapat melihat estimasi waktu pengiriman, berdasarkan lokasi dan ketersediaan layanan.

1.2.Domain Object

Bagian ini menguraikan domain object yang berperan penting dalam sistem informasi pelayanan katering. Domain object ini merepresentasikan entitas atau konsep utama dalam bisnis katering, seperti menu, pesanan, pelanggan, dan lainnya.

Alamat Pengiriman	Kontrol Akses Multi-Level	Pembatalan
Aplikasi Mobile	Konfirmasi	Penghapusan
Cadangan Data	Laporan Penjualan	Pesanan
Customer Service	Laporan Transaksi	Potongan Harga
Dashboard Admin	Live Chat	Rating Pesanan
Database	Menu	Rekomendasi Menu
Estimasi Pengiriman	Metode Pembayaran	Riwayat Pemesanan
Informasi Pembayaran	Mini-Katalog	Status Pesanan
Jadwal Berulang	Mitra	Stok Bahan Makanan
Jadwal Pemesanan	Notifikasi	Transaksi Checkout
Keamanan Data	Keranjang Belanja	Ulasan Pesanan

2. Kebutuhan Fungsional dan Non-Fungsional

Berikut adalah daftar kebutuhan Fungsional dan Non-Fungsional untuk **sistem informasi pelayanan katering**.

Update

2.1 Pemetaan Kebutuhan Fungsional

Berikut adalah daftar kebutuhan Fungsional untuk **sistem informasi pelayanan katering**:

Kode KF	Nama KF	Kode Use Case	Nama Use Case
KF1	Manajemen Menu	UC1.1	Create Data Menu
		UC1.2	Read Data Menu
		UC1.3	Update Data Menu
		UC1.4	Delete Data Menu
KF2	Manajemen Pemesanan	UC2.1	Create Data Pemesanan
		UC2.2	Read Data Pemesanan
		UC2.3	Update Data Pemesanan
		UC2.4	Delete Data Pemesanan
KF3	Manajemen Pembayaran	UC2.1	Create Data Pembayaran
		UC2.2	Read Data Pembayaran
		UC2.3	Update Data Pembayaran
		UC2.4	Delete Data Pembayaran
KF4	Manajemen Pengiriman	UC2.1	Create Data Pengiriman
		UC2.2	Read Data Pengiriman
		UC2.3	Update Data Pengiriman
		UC2.4	Delete Data Pengiriman

Penjelasan :

1.1 Manajemen Menu

1. Sistem dapat menambahkan menu katering baru beserta nama, deskripsi, harga, dan foto makanan.
2. Sistem dapat melihat daftar menu yang tersedia sebelum melakukan pemesanan.
3. Sistem dapat memperbarui informasi menu, seperti harga atau ketersediaan.
4. Sistem dapat menghapus menu yang sudah tidak tersedia atau tidak digunakan.

1.2 Manajemen Pemesanan

1. Sistem dapat memilih menu dan jumlah porsi, lalu melakukan pemesanan.
2. Sistem dapat mengecek status pesanan, apakah sedang diproses, dalam pengiriman, atau telah diterima.

3. Sistem dapat melakukan perubahan pada pesanan sebelum batas waktu tertentu.
4. Sistem dapat membatalkan pesanan sesuai dengan kebijakan yang berlaku.

1.3 Manajemen Pembayaran

1. Sistem dapat membayar pesanan menggunakan berbagai metode pembayaran seperti transfer bank, e-wallet, atau kartu kredit.
2. Sistem dapat melihat daftar transaksi pembayaran yang telah dilakukan.
3. Sistem dapat secara otomatis atau manual dapat mengonfirmasi pembayaran sebelum pesanan diproses lebih lanjut.

1.4 Manajemen Pengiriman

1. Sistem dapat mencatat dan mengatur jadwal pengiriman berdasarkan waktu yang dipilih pelanggan.
2. Sistem dapat memantau status pengiriman pesanan mereka secara real-time.
3. Sistem dapat mengonfirmasi bahwa pesanan telah diterima dengan baik.

2.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional mencakup aspek-aspek yang tidak terkait langsung dengan fungsi sistem tetapi penting untuk memastikan kinerja, keamanan, dan keandalan sistem informasi pelayanan catering.

1. Keamanan (Security)

- Sistem harus menerapkan enkripsi data untuk melindungi informasi pelanggan, termasuk informasi pembayaran.
- Harus ada sistem autentikasi dan otorisasi berbasis peran (role-based access control) untuk membatasi akses ke fitur tertentu.
- Sistem harus memiliki mekanisme proteksi terhadap serangan siber seperti SQL Injection, XSS (Cross-Site Scripting), dan CSRF (Cross-Site Request Forgery).

2. Kinerja (Performance)

- Sistem harus dapat menangani minimal 500 pengguna secara bersamaan tanpa mengalami penurunan kinerja yang signifikan.
- Waktu respon sistem tidak boleh melebihi 10 detik untuk pemrosesan pesanan dan transaksi pembayaran.
- Sistem harus mampu menangani minimal 50.000 transaksi per bulan

3. Keandalan (Reliability)

- Sistem harus memiliki uptime minimal 99,5% dalam satu bulan.
- Sistem harus dapat menangani pemulihan data otomatis jika terjadi kegagalan server.
- Harus ada mekanisme backup data harian dan pemulihan data dalam waktu maksimal 24 jam setelah kegagalan terjadi.

4. Usability (Kemudahan Penggunaan)

- Antarmuka pengguna harus responsif dan dapat digunakan di berbagai perangkat (desktop, tablet, dan mobile).
- Sistem harus memiliki navigasi yang intuitif dengan dokumentasi panduan bagi pengguna.
- Harus ada fitur bantuan atau FAQ yang dapat diakses langsung oleh pengguna.

5. Kompatibilitas (Compatibility)

- Sistem harus kompatibel dengan browser modern seperti Google Chrome, Mozilla Firefox, Safari, dan Microsoft Edge.
- Sistem harus dapat berjalan di berbagai sistem operasi termasuk Windows, macOS, Android, dan iOS.

6. Scalability (Skalabilitas)

- Sistem harus dirancang dengan arsitektur yang memungkinkan ekspansi jumlah pengguna dan fitur di masa mendatang.
- Sistem harus mampu beradaptasi dengan peningkatan volume transaksi dan jumlah data yang besar.

7. Maintainability (Kemudahan Pemeliharaan)

- Kode sumber harus didokumentasikan dengan baik agar mudah diperbaiki dan dikembangkan lebih lanjut.
- Sistem harus memiliki fitur logging untuk mendeteksi dan menganalisis kesalahan dengan mudah.
- Harus ada mekanisme pemantauan sistem untuk mengidentifikasi potensi masalah sebelum terjadi gangguan.

8. Portability (Portabilitas)

- Sistem harus dapat dipindahkan ke lingkungan hosting yang berbeda tanpa memerlukan perubahan besar.
- Sistem harus mendukung deployment di lingkungan cloud seperti AWS, Google Cloud, atau Azure.

9. Legal Compliance (Kepatuhan Hukum)

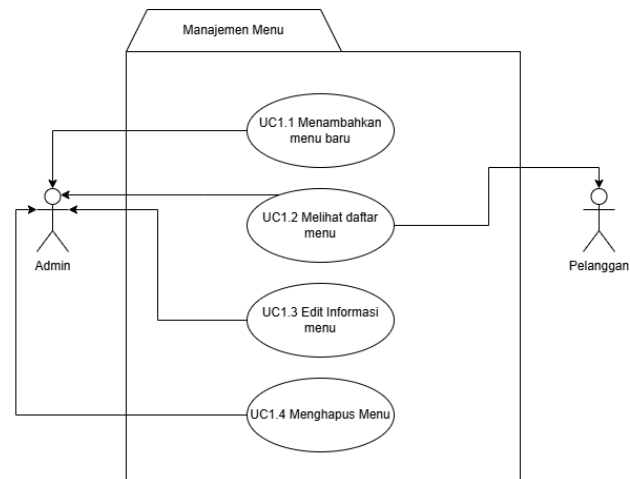
- Sistem harus mematuhi regulasi perlindungan data pelanggan seperti GDPR atau aturan lokal yang berlaku.
- Sistem harus memiliki kebijakan privasi yang transparan dan dapat diakses oleh pengguna.

Dengan memenuhi kebutuhan non-fungsional ini, sistem informasi pelayanan katering dapat beroperasi dengan efisien, aman, dan dapat diandalkan sesuai dengan standar industri.

3. Use Case Diagram

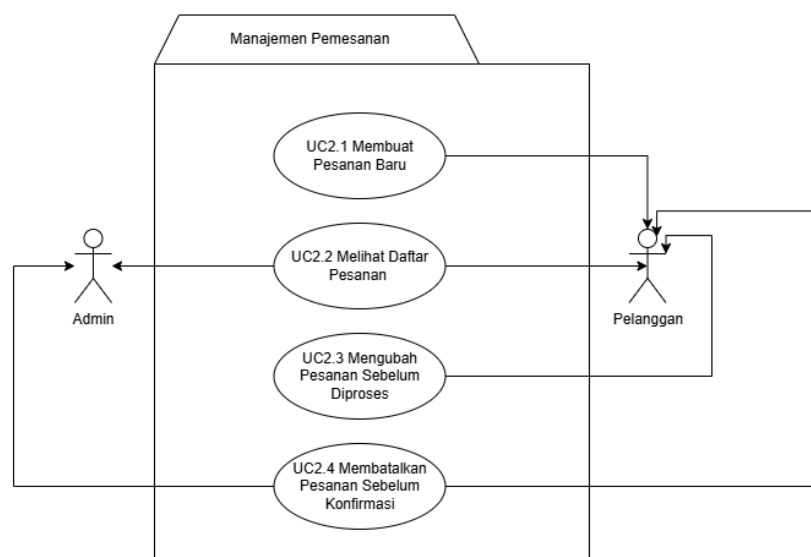
3.1 Use Case Diagram Manajemen Menu

Berikut adalah hasil *use case diagram* dari proses menu katering. *Use case diagram* ini dirancang berdasarkan paradigma CRUD (*Create, Read, Update, Delete*) untuk memastikan setiap data dalam sistem dapat dikelola dengan baik. Diagram ini membantu dalam memahami batasan akses setiap aktor serta bagaimana mereka berinteraksi dengan fitur-fitur sistem katering.



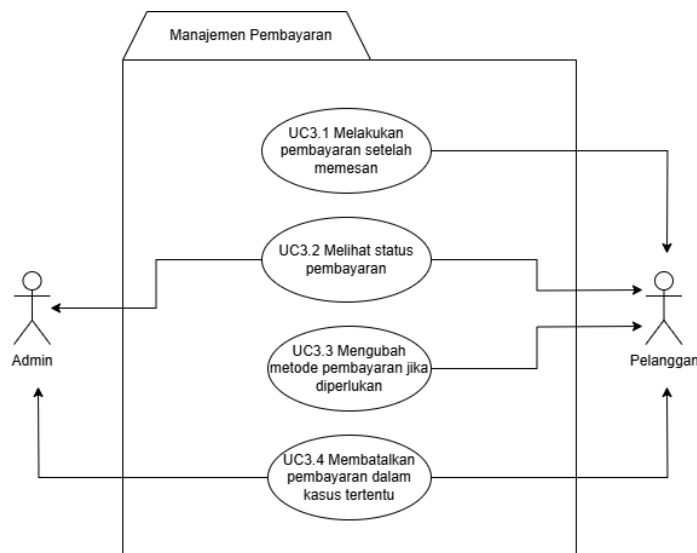
3.2 Use Case Diagram Manajemen Pemesanan

Berikut adalah hasil *use case diagram* dari proses pemesanan katering. *Use case diagram* ini dirancang berdasarkan paradigma CRUD (*Create, Read, Update, Delete*) untuk memastikan setiap data dalam sistem dapat dikelola dengan baik. Diagram ini membantu dalam memahami batasan akses setiap aktor serta bagaimana mereka berinteraksi dengan fitur-fitur sistem katering.



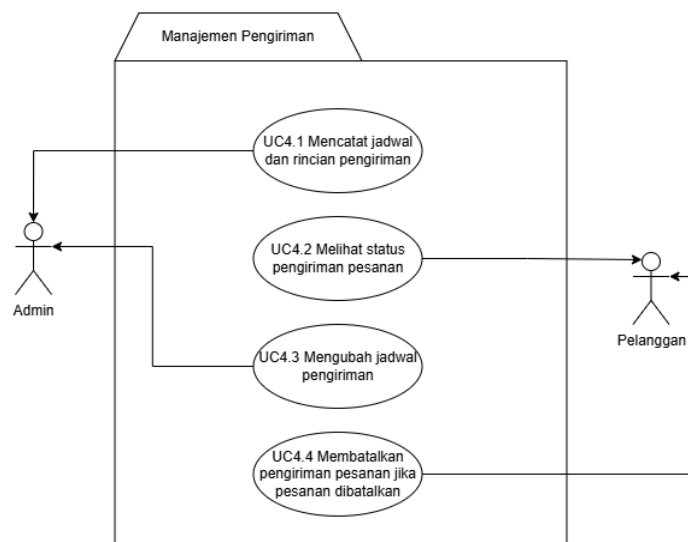
3.3 Use Case Diagram Manajemen Pembayaran

Berikut adalah hasil *use case diagram* dari proses pembayaran catering. *Use case diagram* ini dirancang berdasarkan paradigma CRUD (*Create, Read, Update, Delete*) untuk memastikan setiap data dalam sistem dapat dikelola dengan baik. Diagram ini membantu dalam memahami batasan akses setiap aktor serta bagaimana mereka berinteraksi dengan fitur-fitur sistem catering.



3.4 Use Case Diagram Manajemen Pengiriman

Berikut adalah hasil *use case diagram* dari proses pengiriman catering. *Use case diagram* ini dirancang berdasarkan paradigma CRUD (*Create, Read, Update, Delete*) untuk memastikan setiap data dalam sistem dapat dikelola dengan baik. Diagram ini membantu dalam memahami batasan akses setiap aktor serta bagaimana mereka berinteraksi dengan fitur-fitur sistem catering.



4. Use Case Narrative

Nama Use Case	Create Data Menu
Tujuan	Memungkinkan admin untuk menambahkan menu baru ke dalam sistem katering, sehingga menu dapat dikelola dan ditampilkan kepada pelanggan.
Aktor	Admin – Bertanggung jawab untuk mengelola data menu di sistem.
Pre-Kondisi	<ol style="list-style-type: none"> 1. Admin telah berhasil login ke dalam sistem. 2. Sistem dalam kondisi aktif dan dapat menerima input. 3. Admin memiliki akses ke fitur manajemen menu.
Skenario Sukses	<ol style="list-style-type: none"> 1. Admin memilih fitur "Tambah Menu". 2. Sistem menampilkan formulir input menu. 3. Admin mengisi data menu (nama, deskripsi, harga, kategori, gambar, dll.). 4. Admin mengklik tombol "Simpan". 5. Sistem memvalidasi data yang diinput. 6. Jika valid, sistem menyimpan data menu ke database. 7. Sistem menampilkan notifikasi bahwa menu berhasil ditambahkan.
Skenario Alternatif	<p>A1: Data tidak valid</p> <ul style="list-style-type: none"> -Sistem mendeteksi kesalahan pada data (misalnya, harga tidak valid atau field wajib kosong). -Sistem menampilkan pesan kesalahan dan meminta admin memperbaiki input. - Admin memperbaiki data dan mengirim ulang. <p>A2: Koneksi terputus</p> <ul style="list-style-type: none"> - Koneksi terputus sebelum data tersimpan. -Sistem menampilkan pesan error dan meminta admin mencoba lagi nanti.

Nama Use Case	Read Data Menu
Tujuan	Memungkinkan admin untuk melihat daftar menu yang telah tersimpan dalam sistem katering.
Aktor	Admin - Bertanggung Jawab untuk mengelola dan mengecek data menu di sistem.
Pre-Kondisi	<ol style="list-style-type: none"> 1. Admin telah login ke dalam sistem. 2. Sistem dalam kondisi aktif. 3. Admin memiliki akses ke fitur manajemen menu.
Skenario Sukses	<ol style="list-style-type: none"> 1. Admin memilih fitur "Lihat Data Menu". 2. Sistem mengambil data menu dari database. 3. Sistem menampilkan daftar menu yang tersedia. 4. Admin dapat melihat informasi menu seperti nama, harga, dan kategori.
Skenario Alternatif	<p>A1: Data tidak tersedia</p> <p>Sistem tidak menemukan data menu.- Sistem menampilkan pesan "Tidak ada menu yang tersedia."</p>

	A2: Koneksi terputus Sistem gagal mengambil data karena koneksi bermasalah.- Sistem menampilkan pesan error dan menyarankan admin mencoba kembali.
--	--

Nama Use Case	Update Data Menu
Tujuan	Memungkinkan admin untuk memperbarui data menu yang sudah ada dalam sistem.
Aktor	Admin- Bertanggung Jawab untuk mengelola update data menu di sistem.
Pre-Kondisi	1. Admin telah login ke sistem. 2. Data menu yang akan diperbarui sudah tersedia dalam sistem.
Skenario Sukses	1. Admin memilih opsi "Update Data Menu" di sistem. 2. Sistem menampilkan daftar menu yang tersedia. 3. Admin memilih data menu yang akan diperbarui. 4. Sistem menampilkan detail data menu yang dipilih. 5. Admin memperbarui informasi menu (misalnya nama, harga, deskripsi). 6. Admin menyimpan perubahan. 7. Sistem memvalidasi data yang dimasukkan. 8. Jika validasi berhasil, sistem menyimpan perubahan ke database. 9. Sistem menampilkan notifikasi bahwa perubahan telah berhasil disimpan.
Skenario Alternatif	A1: Jika validasi gagal: 1. Sistem menampilkan pesan kesalahan. 2. Admin dapat memperbaiki data dan mencoba menyimpan kembali. A2: Jika admin membatalkan proses pembaruan: 1. Sistem membatalkan perubahan dan kembali ke tampilan sebelumnya.

Nama Use Case	Delete Data Menu
Tujuan	Memungkinkan admin untuk menghapus menu catering yang sudah tidak tersedia atau tidak digunakan dari sistem.
Aktor	Admin - Bertanggung jawab untuk menghapus menu yang sudah tidak tersedia.
Pre-Kondisi	1. Admin sudah login ke dalam sistem. 2. Data menu yang akan dihapus sudah tersedia dalam sistem.
Skenario Sukses	1. Admin mengakses halaman manajemen menu. 2. Admin memilih menu yang ingin dihapus. 3. Sistem menampilkan detail menu dan konfirmasi penghapusan. 4. Admin mengkonfirmasi penghapusan. 5. Sistem menghapus data menu dari database. 6. Sistem memberikan notifikasi bahwa menu berhasil dihapus.
Skenario Alternatif	A1: Menu sedang digunakan dalam pesanan aktif Jika menu yang ingin dihapus sedang digunakan dalam pesanan aktif, sistem menampilkan peringatan dan menolak penghapusan. A2: Koneksi terputus atau terjadi kesalahan sistem

	Jika terjadi kesalahan sistem atau koneksi terputus, sistem menampilkan pesan error dan meminta admin mencoba kembali.
--	--

Nama Use Case	Create Data Pemesanan
Tujuan	Memungkinkan admin untuk menambahkan data menu baru ke dalam sistem.
Aktor	Admin-Bertanggung jawab untuk mengelola data Pemesanan di sistem.
Pre-Kondisi	1. Admin telah login ke sistem. 2. Sistem dalam keadaan aktif dan dapat menerima input.
Skenario Sukses	1. Admin memilih fitur "Tambah Menu". 2. Sistem menampilkan form input untuk data menu. 3. Admin mengisi data menu (nama menu, kategori, harga, deskripsi, dll.). 4. Admin mengklik tombol "Simpan". 5. Sistem memvalidasi data yang dimasukkan. 6. Jika valid, sistem menyimpan data menu ke dalam database. 7. Sistem menampilkan pesan konfirmasi bahwa data menu berhasil disimpan.
Skenario Alternatif	A1: Data tidak valid 1. Jika ada data yang tidak valid (misalnya harga tidak berupa angka), sistem menampilkan pesan error. 2. Admin memperbaiki data dan mengulangi proses penyimpanan. A2: Koneksi database gagal 1. Jika terjadi kegagalan koneksi database, sistem menampilkan pesan error dan meminta admin mencoba kembali nanti.

Nama Use Case	Read Data Pemesanan
Tujuan	Memungkinkan aktor untuk melihat dan membaca data pemesanan yang tersimpan dalam sistem.
Aktor	Admin, Pelanggan
Pre-Kondisi	1. Sistem sudah memiliki data pemesanan yang tersimpan. 2. Aktor telah login ke sistem dengan hak akses yang sesuai.
Skenario Sukses	1. Aktor masuk ke sistem dan memilih menu "Pemesanan". 2. Sistem menampilkan daftar pemesanan yang tersedia. 3. Aktor dapat melihat detail pemesanan tertentu dengan memilih salah satu pemesanan. 4. Sistem menampilkan informasi lengkap dari pemesanan yang dipilih.
Skenario Alternatif	A1: Jika tidak ada data pemesanan, sistem menampilkan pesan "Tidak ada data pemesanan yang tersedia". A2: Jika terjadi kegagalan koneksi ke database, sistem menampilkan pesan error dan meminta pengguna untuk mencoba kembali.

Nama Use Case	Update Data Pemesanan
Tujuan	Memungkinkan pengguna untuk memperbarui data pemesanan yang sudah ada di sistem.
Aktor	Admin atau staf yang bertanggung jawab atas manajemen pemesanan.
Pre-Kondisi	1. Pengguna telah berhasil login ke sistem. 2. Data pemesanan yang akan diperbarui sudah tersedia di database.
Skenario Sukses	1. Pengguna memilih menu "Manajemen Pemesanan". 2. Pengguna mencari data pemesanan yang ingin diperbarui menggunakan fitur pencarian. 3. Sistem menampilkan detail data pemesanan yang dipilih. 4. Pengguna memperbarui informasi sesuai kebutuhan (misalnya, jumlah pesanan, tanggal pengiriman, dll.). 5. Pengguna menyimpan perubahan. 6. Sistem memvalidasi data dan memperbarui informasi di database. 7. Sistem menampilkan pesan konfirmasi bahwa data berhasil diperbarui
Skenario Alternatif	1a. Jika data pemesanan tidak ditemukan: - Sistem menampilkan pesan kesalahan "Data tidak ditemukan". - Pengguna dapat mencoba pencarian ulang atau kembali ke menu utama. 4a. Jika data yang diinput tidak valid: - Sistem menampilkan pesan kesalahan terkait validasi (misalnya, format tanggal salah). - Pengguna memperbaiki data dan mengulangi proses penyimpanan

Nama Use Case	
Tujuan	
Aktor	
Pre-Kondisi	
Skenario Sukses	
Skenario Alternatif	

Nama Use Case	
Tujuan	
Aktor	
Pre-Kondisi	
Skenario Sukses	
Skenario Alternatif	

Nama Use Case	
Tujuan	
Aktor	
Pre-Kondisi	
Skenario Sukses	

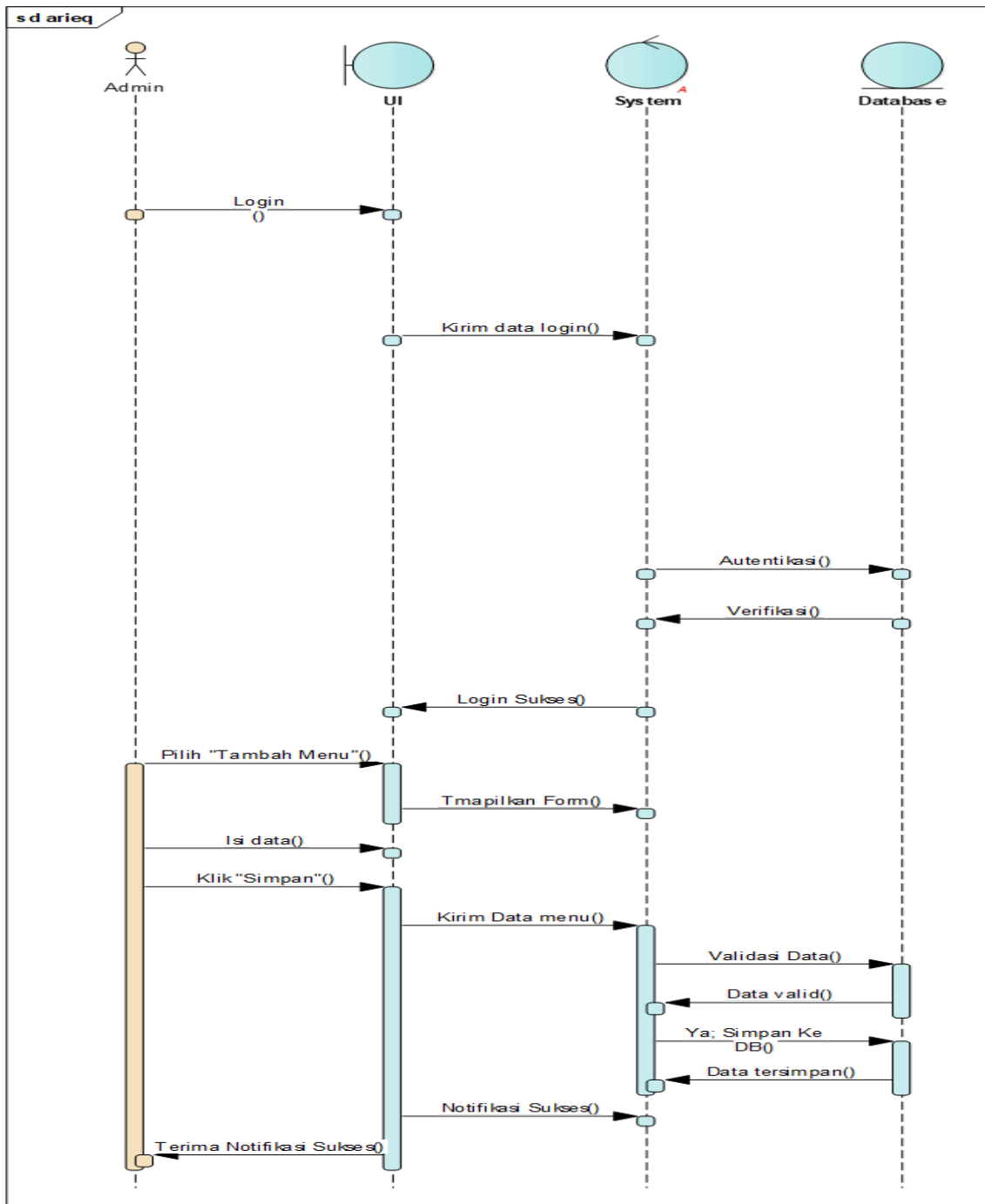
Skenario Alternatif	
------------------------	--

5. Sequence Diagram

1 use case narrative = 1 sequence diagram

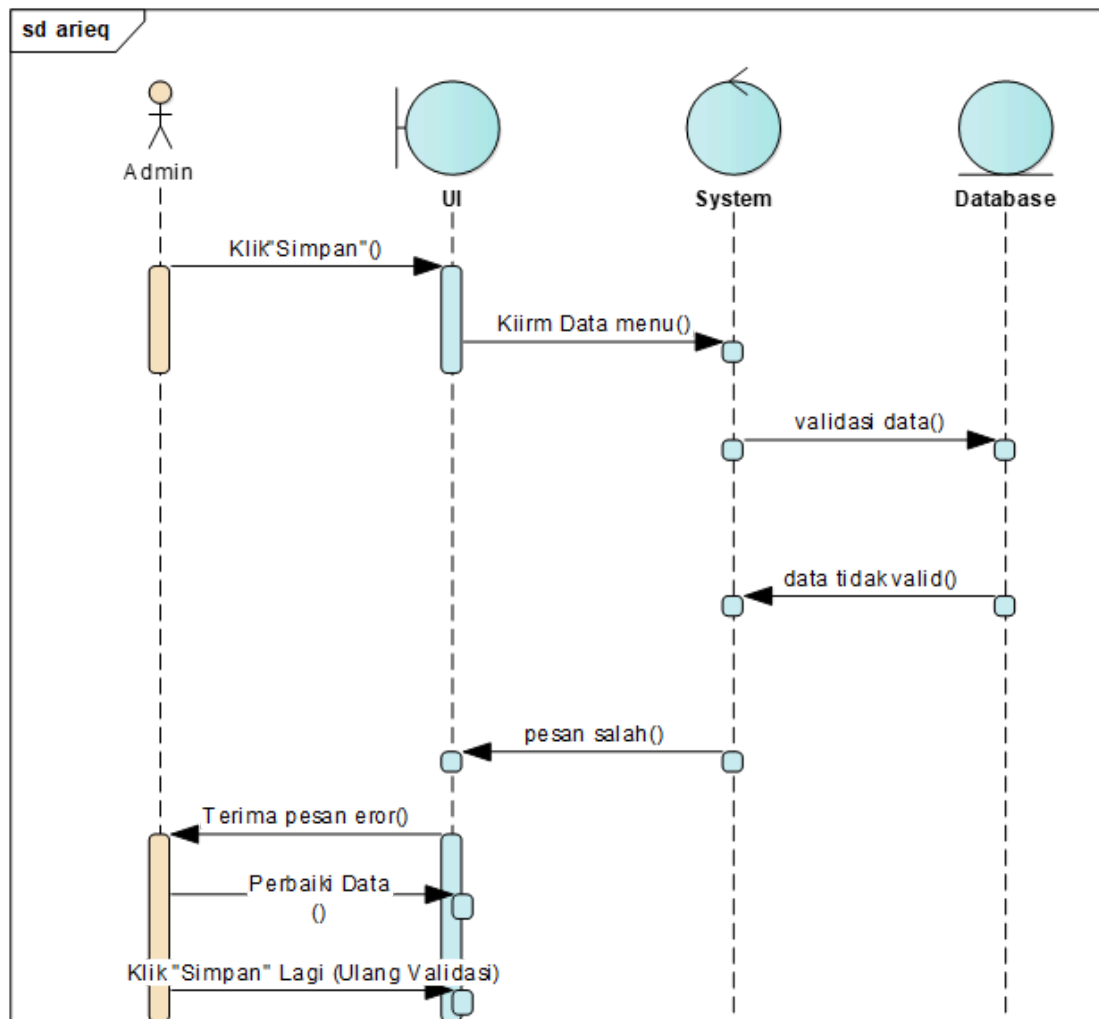
1. Use Case Narrative

- Skenario Sukses

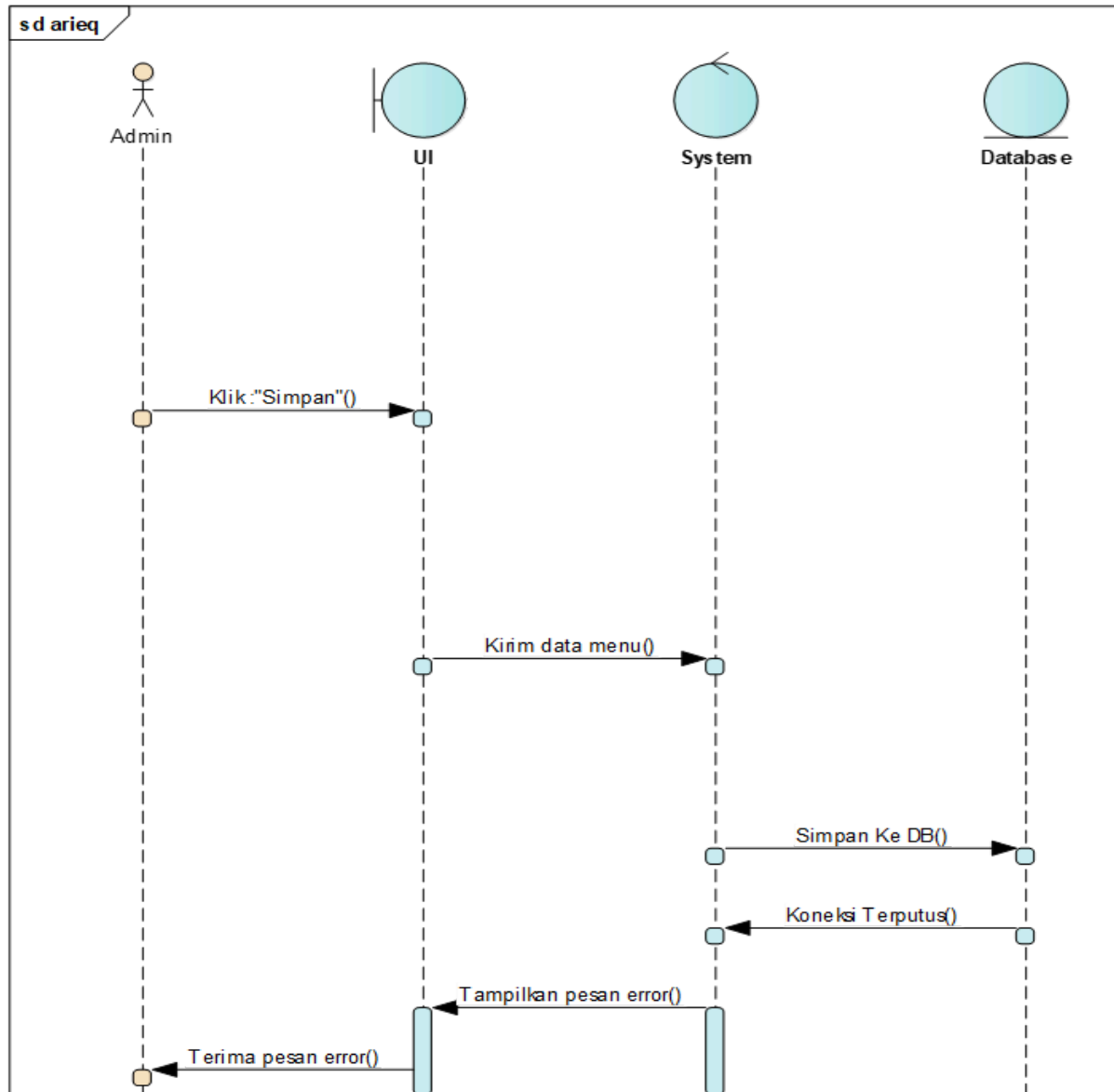


- Skenario Alternatif

A1: Data Tidak Valid

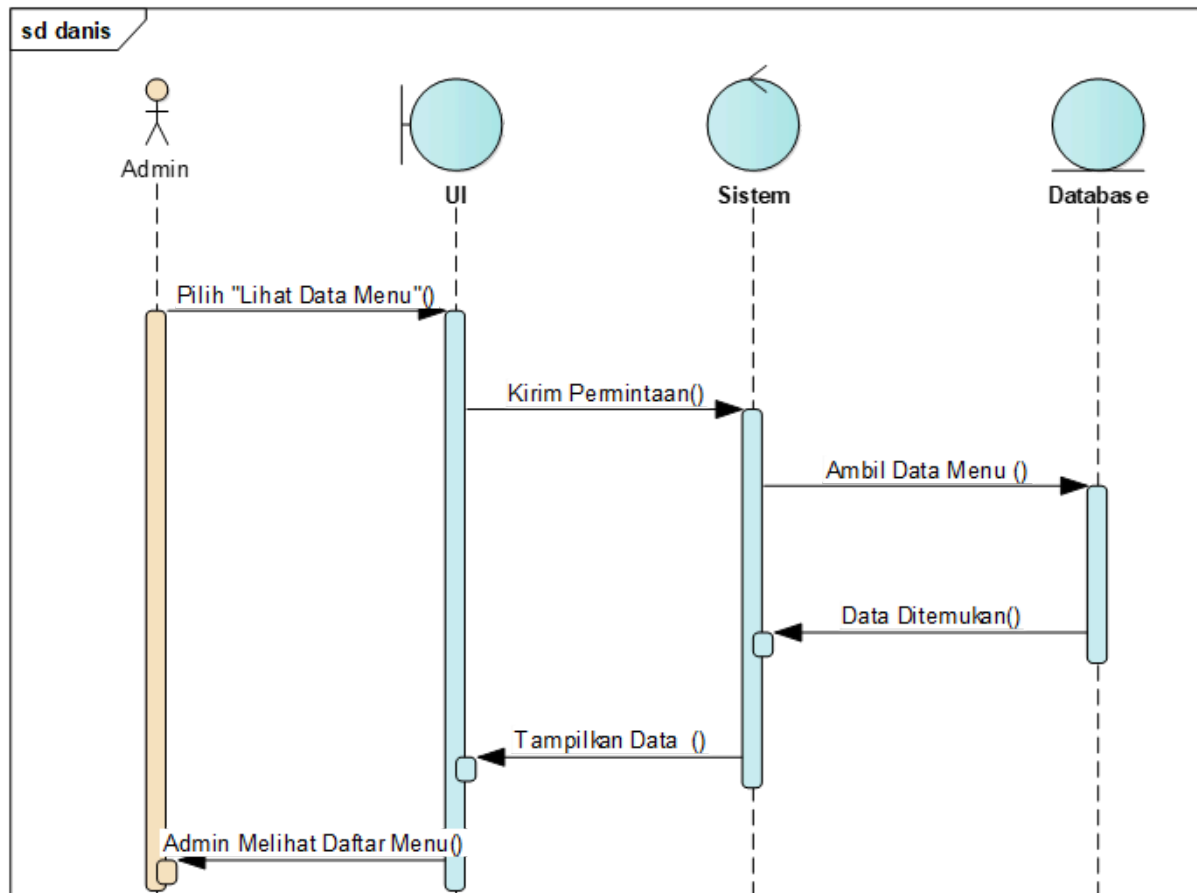


A2: Koneksi Terputus



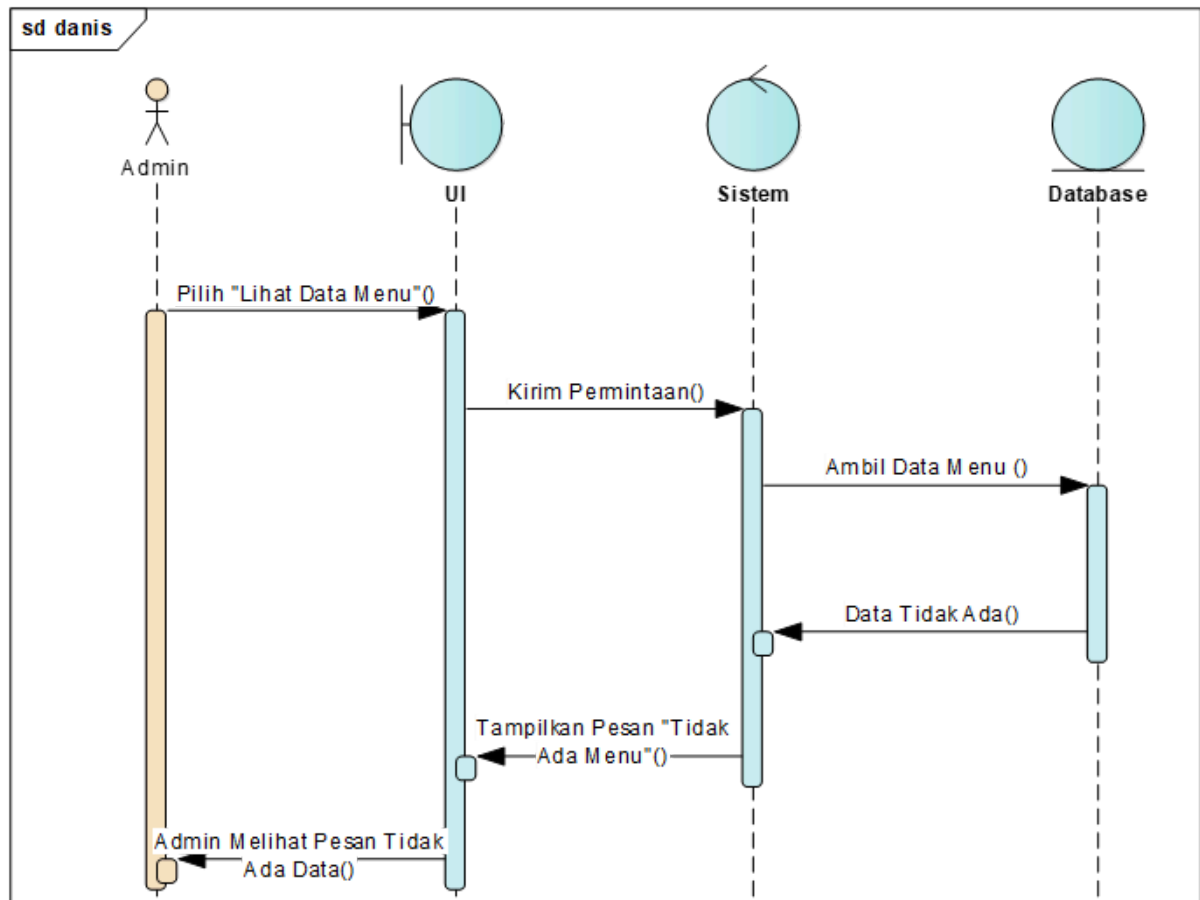
2. Use Case Narrative

- Skenario Sukses

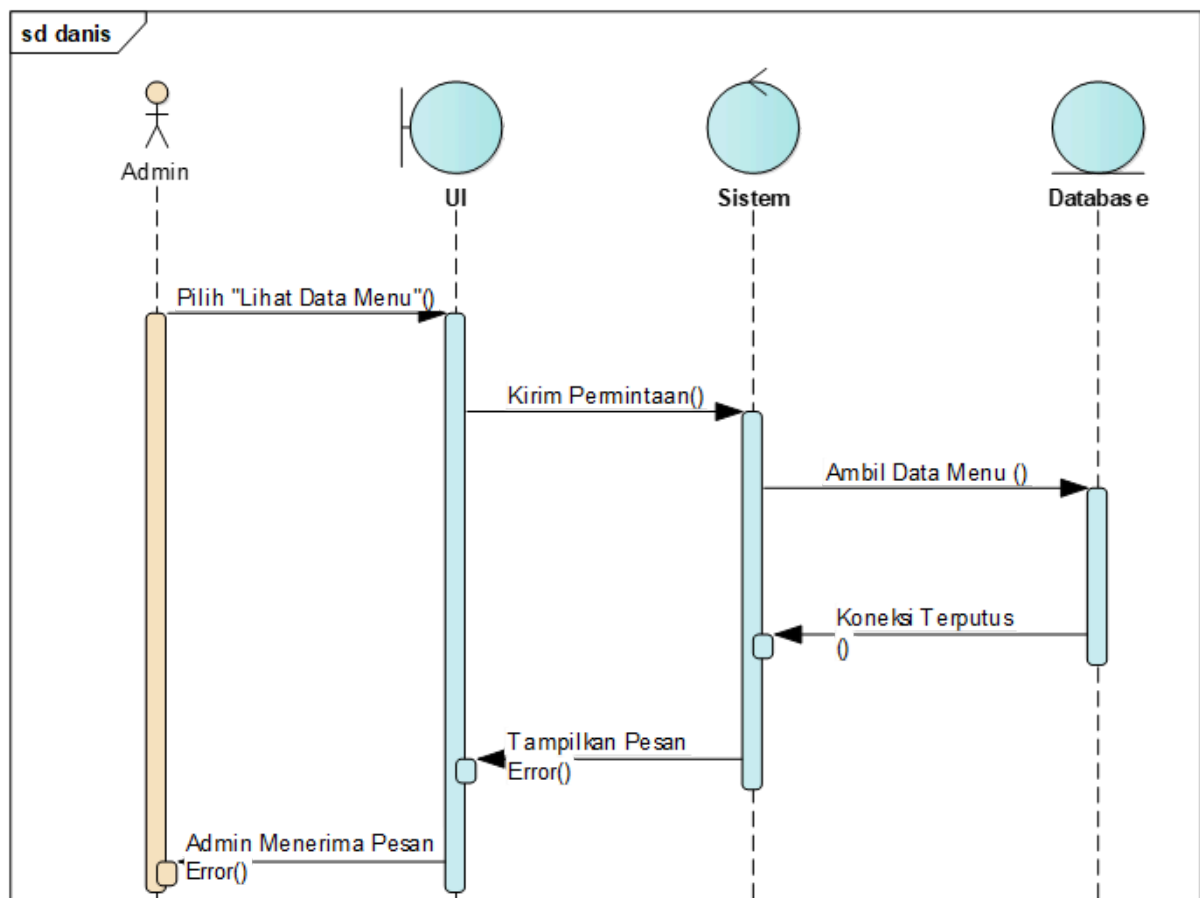


- Skenario Alternatif

A1: Data Tidak Tersedia

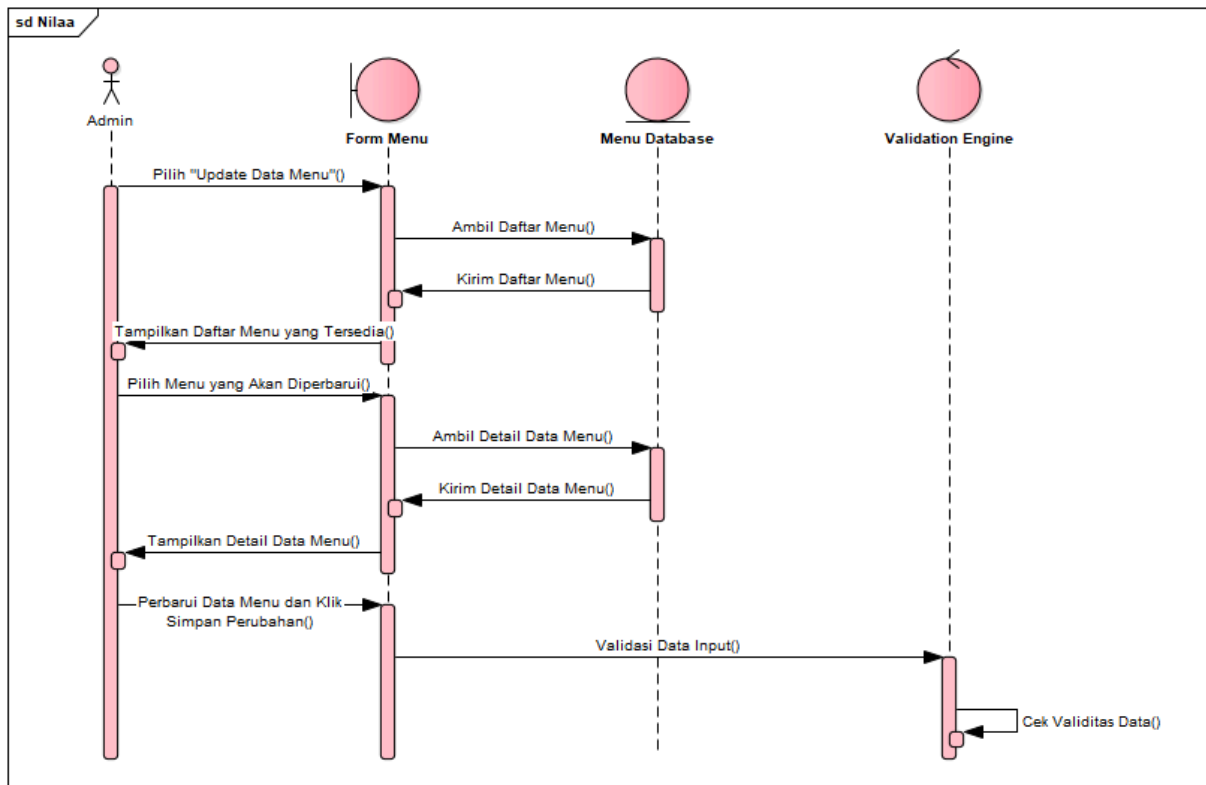


A2: Koneksi Terputus



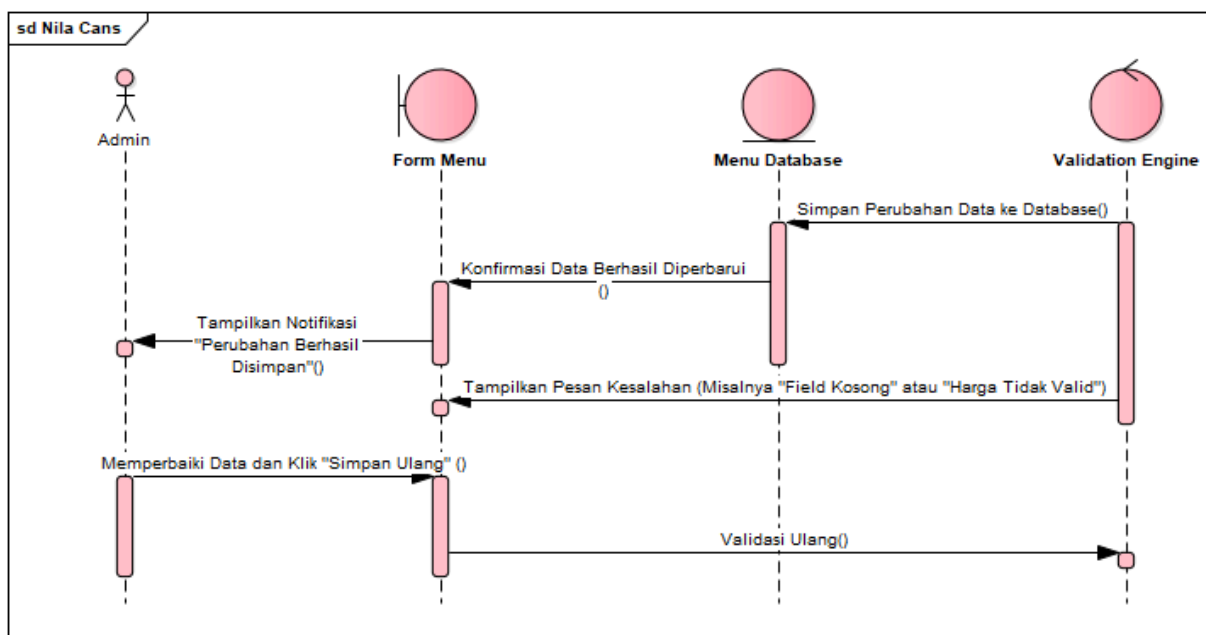
3. Use Case Narrative Update Data Menu

- Skenario Sukses



- **Skenario Alternatif**

A1 : Validasi Gagal



A2 : Admin Membatalkan Proses Pembaruan

