

BAB III

ANALISIS DAN PERANCANGAN

3.1. Analisis Permasalahan

Perusahaan saat ini hanya menggunakan Excel untuk visualisasi data, yang kurang interaktif dan sulit diakses di berbagai perangkat. Hal ini membatasi kemampuan manajemen untuk memantau dan menganalisis data secara cepat. Selain itu, aksesibilitas data menjadi kendala signifikan. Hal ini menghambat kemampuan manajemen untuk melakukan pengambilan keputusan yang cepat dan tepat berdasarkan data yang tersedia.

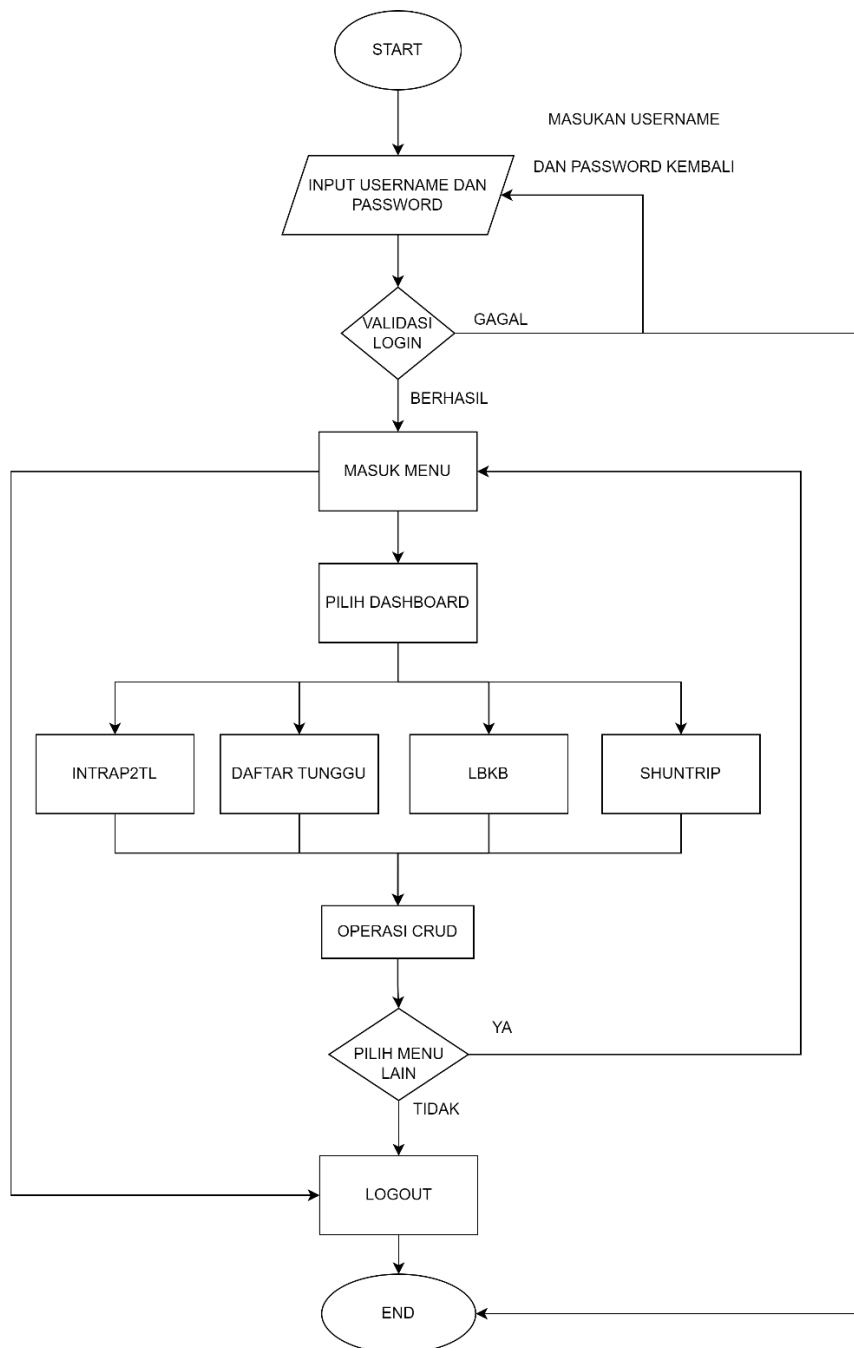
3.2. Usulan Pemecahan Masalah

Untuk mengatasi masalah ini, diusulkan pengembangan dashboard web interaktif dan responsif menggunakan teknologi HTML, CSS, PHP, dan JavaScript. Dashboard ini akan memungkinkan akses data cepat dan mempermudah pengguna memantau dan menganalisis data. Desain antarmuka yang intuitif akan membantu pengguna yang sebelumnya terbiasa dengan Excel untuk beradaptasi dengan cepat. Agar mempermudah analisis di berbagai perangkat.

3.3. Desain Sistem

Desain sistem adalah proses mendefinisikan arsitektur, dan data untuk sistem yang memenuhi persyaratan tertentu. Ini melibatkan penciptaan spesifikasi rinci untuk komponen sistem dan kerangka kerja yang memungkinkan mereka bekerja bersama dengan efisien. Tujuan utamanya adalah untuk memastikan bahwa sistem akhir beroperasi dengan lancar, memenuhi kebutuhan pengguna, dan dapat diadaptasi untuk perubahan masa depan. Desain sistem yang baik memerlukan pemahaman yang mendalam tentang masalah yang dihadapi, serta kreativitas dan inovasi untuk menemukan solusi yang optimal. Ini sering melibatkan penggunaan bahasa pemodelan untuk menggambarkan interaksi antara berbagai komponen sistem dan memastikan bahwa semua bagian berfungsi bersama.

3.3.1. Penggambaran *FlowChart*



Gambar 3.1 *Flowchart*

Dari *Flowchart* di atas akan dijelaskan beberapa proses yang terjadi yaitu :

1. Mulai: Proses dimulai dengan input username dan password.
2. Validasi: Sistem memeriksa kebenaran data yang dimasukkan.
3. Login Gagal: Jika validasi tidak berhasil, proses berakhir.
4. Pilih Dashboard: Jika validasi berhasil, pengguna memilih dashboard.
5. Admin Check: Sistem memeriksa apakah pengguna adalah admin.
6. Operasi Admin: Jika pengguna adalah admin, mereka dapat membuat, mengupdate, atau menghapus data.
7. Melihat Chart: Pengguna dapat melihat chart atau informasi pada dashboard.
8. Logout: Setelah selesai, pengguna akan logout dari sistem.

3.3.2. Analisis Software

Dalam pembuatan aplikasi inventori Situs ini, terdapat beberapa software yang dibutuhkan, antara lain:

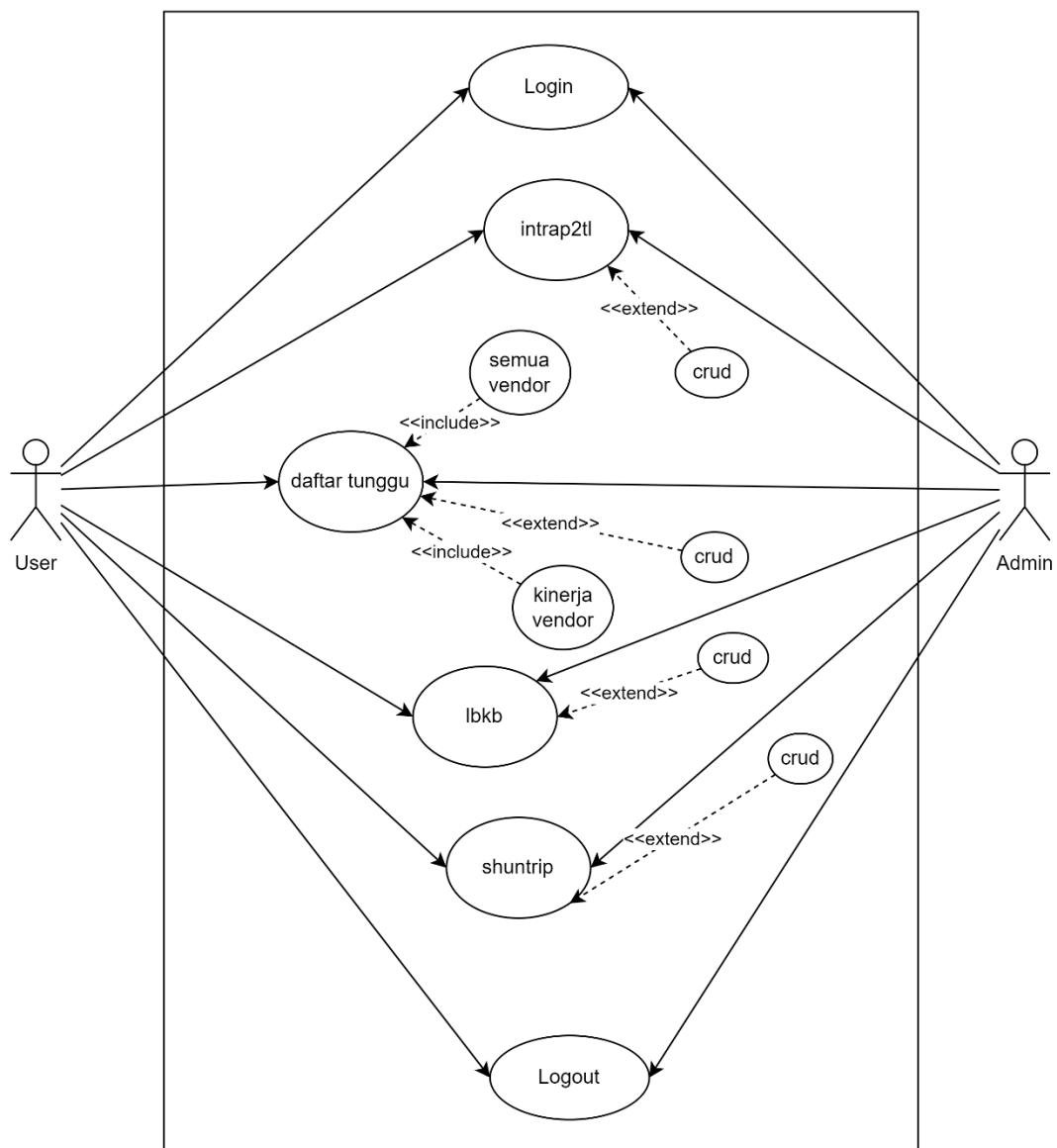
1. Xampp versi 8.2.12
2. MySql
3. PHP versi 8.2

3.3.3. Perancangan Sistem

Perancangan system yang dibuat dalam menentukan kebutuhan system ini sesuai dengan analisis dan perancangan yang digunakan.

3.3.3.1. Use Case Diagram

Use case diagram adalah jenis diagram UML (Unified Modeling Language) yang menggambarkan fungsi, ruang lingkup, dan interaksi pengguna dengan sistem. Diagram ini memvisualisasikan interaksi antara pengguna (aktor) dan sistem (*use case*).



Gambar 3.2 Use case Diagram

Penjelasan dari gambar tersebut antara lain:

Aktor: Terdapat dua aktor, yaitu pengguna dan admin.

Use Case Pengguna:

1. Login: Pengguna memulai interaksi dengan sistem.
2. Melihat Intrap2tl: Pengguna dapat melihat informasi intrap2tl.
3. Melihat Daftar Tunggu: Pengguna dapat melihat daftar tunggu.
4. Melihat LBKB: Pengguna dapat melihat lkbk.
5. Melihat Shuntrip: Pengguna dapat melihat shuntrip.
6. Logout: Pengguna mengakhiri sesi mereka.

Use Case Admin:

1. Manajemen Data: Admin dapat mengelola data.
2. Membuat Data: Admin dapat membuat data baru.
3. Mengubah Data: Admin dapat mengubah data yang ada.
4. Menghapus Data: Admin dapat menghapus data.

Relasi Extend: Tanda "-extend->" menunjukkan fungsionalitas tambahan atau opsi yang tersedia di luar *use case* utama.

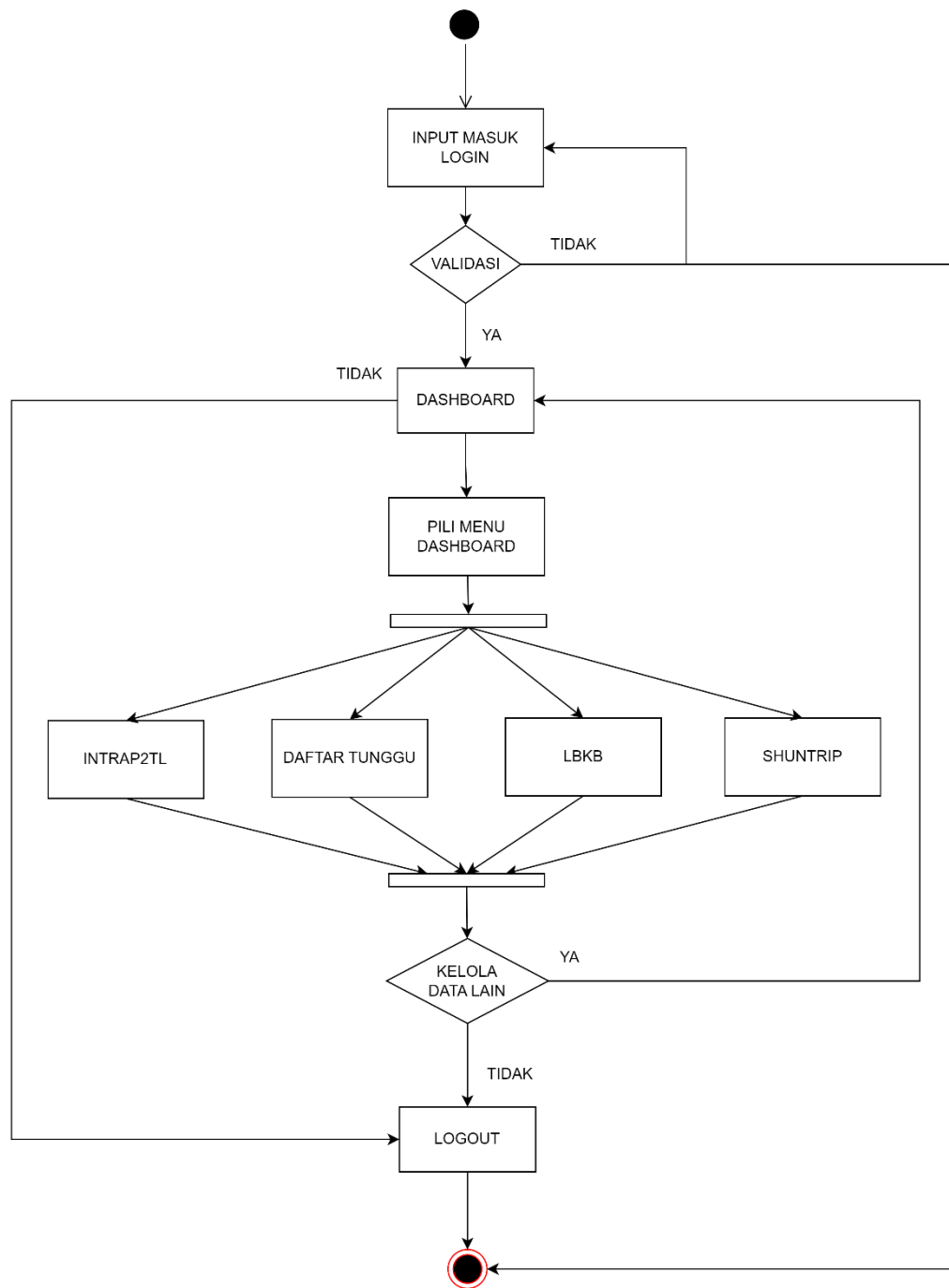
Relasi Include : Tanda "-include ->" menunjukan bagian kecil dari interaksi tersebut.

3.3.3.2. *Activity* Diagram

Activity Diagram adalah bentuk visual yang menggambarkan alur kerja dalam sebuah sistem. Diagram ini berisi urutan proses atau aktivitas yang terjadi, serta hubungan antara langkah-langkah tersebut.

Perbedaan dengan *Flowchart*:

1. *Activity* diagram lebih menggambarkan alur kerja dari awal hingga akhir suatu proses bisnis dalam sistem.
2. Meskipun terlihat seperti *flowchart*, *activity* diagram berbeda karena menggambarkan aktivitas sistem, bukan apa yang dilakukan oleh aktor.



Gambar 3.3 Activity Diagram

Penjelasan dari gambar tersebut adalah sebagai berikut:

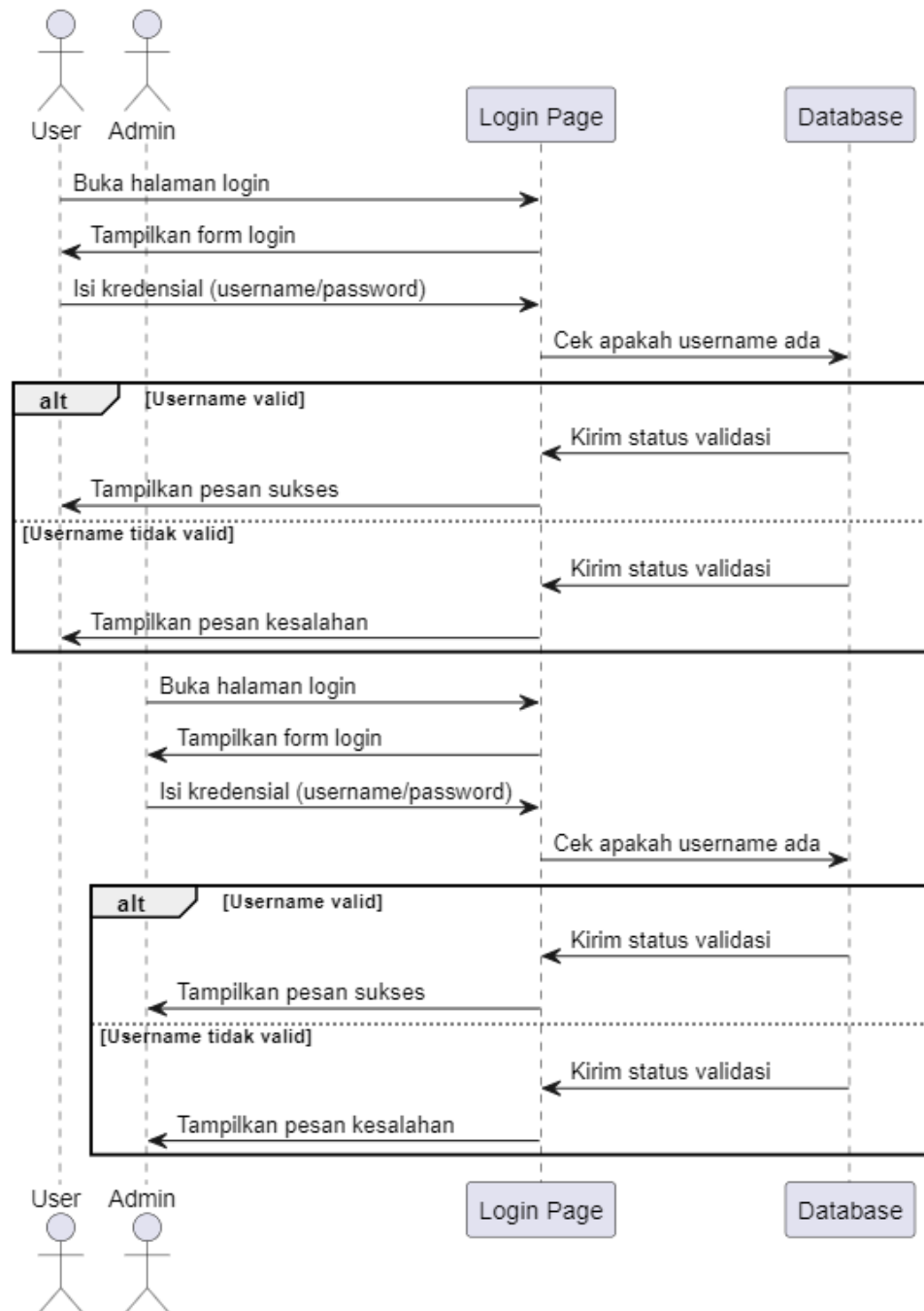
- Input Masuk Login: Pengguna memulai dengan memasukkan informasi login.

- Validasi: Jika validasi berhasil, pengguna melanjutkan ke *dashboard*; jika tidak, hasilnya adalah percobaan login yang gagal.
- Dashboard: Pengguna dapat memilih berbagai opsi menu termasuk "INTRAP2TL," "DAFTAR TUNGGU," "LBKB," dan "SHUNTRIP."
- Kelola Data Lain: Ada juga opsi untuk mengelola data lain jika diperlukan sebelum logout.

3.3.3.3. Sequence diagram

Sequence diagram adalah bentuk visual yang menggambarkan urutan interaksi antara objek-objek dalam sebuah sistem. Diagram ini menunjukkan bagaimana objek-objek berkomunikasi satu sama lain melalui pesan-pesan yang dikirim. Setiap objek memiliki lifeline yang menunjukkan waktu aktifnya selama proses berlangsung. Pesan-pesan antara objek-objek digambarkan dengan panah yang mengindikasikan arah komunikasi. *Sequence* diagram membantu memvisualisasikan alur kerja sistem dan memahami bagaimana proses berlangsung secara keseluruhan.

1. Sequence diagram login



Gambar 3.4 Sequence Diagram Login

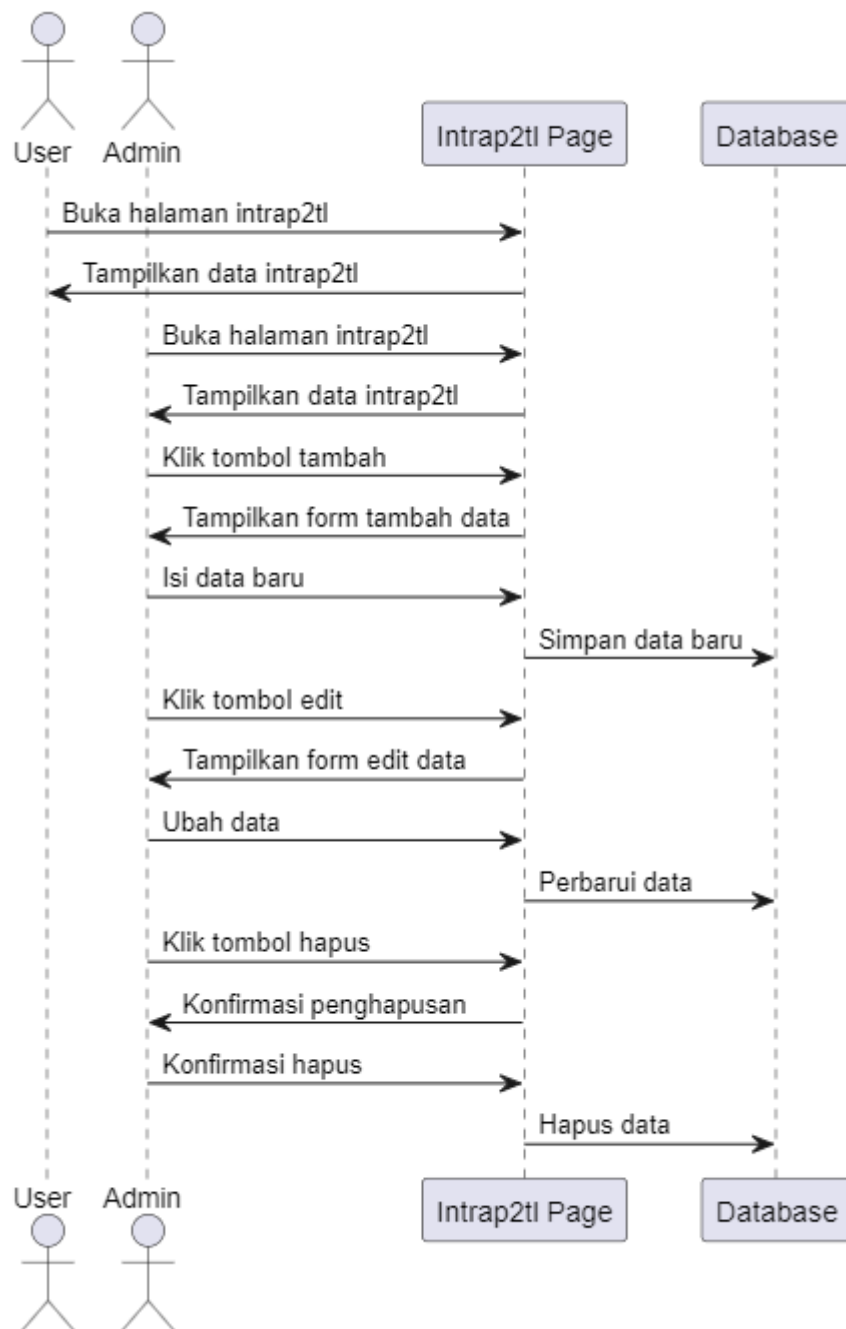
Penjelasan dari gambar tersebut adalah sebagai berikut:

1. Masukan Kredensial: Pengguna atau admin memulai dengan memasukkan username dan password.
2. Pemeriksaan Kredensial: *Login Page* memeriksa kredensial yang dimasukkan.
3. Hasil Pemeriksaan: *Database* mengembalikan hasil pemeriksaan.
4. Pesan Berhasil/Gagal:
5. Jika berhasil, pesan sukses ditampilkan dan pengguna/admin diarahkan ke *Home Page* masing-masing.
6. Jika gagal, pesan error ditampilkan.

Admin memiliki opsi tambahan untuk mengedit, melihat, atau menghapus data. Diagram ini membantu memvisualisasikan proses otentikasi dan otorisasi dalam sistem serta tindakan yang dapat dilakukan oleh pengguna dan admin setelah login berhasil.

2. Sequence diagram intrap2tl

Proses dimulai ketika Admin berhasil masuk ke *Dashboard* Aplikasi. Admin dan user dapat memilih menu intrap2tl, kemudian admin dapat menambah, mengubah, menghapus, data dari menu tersebut.

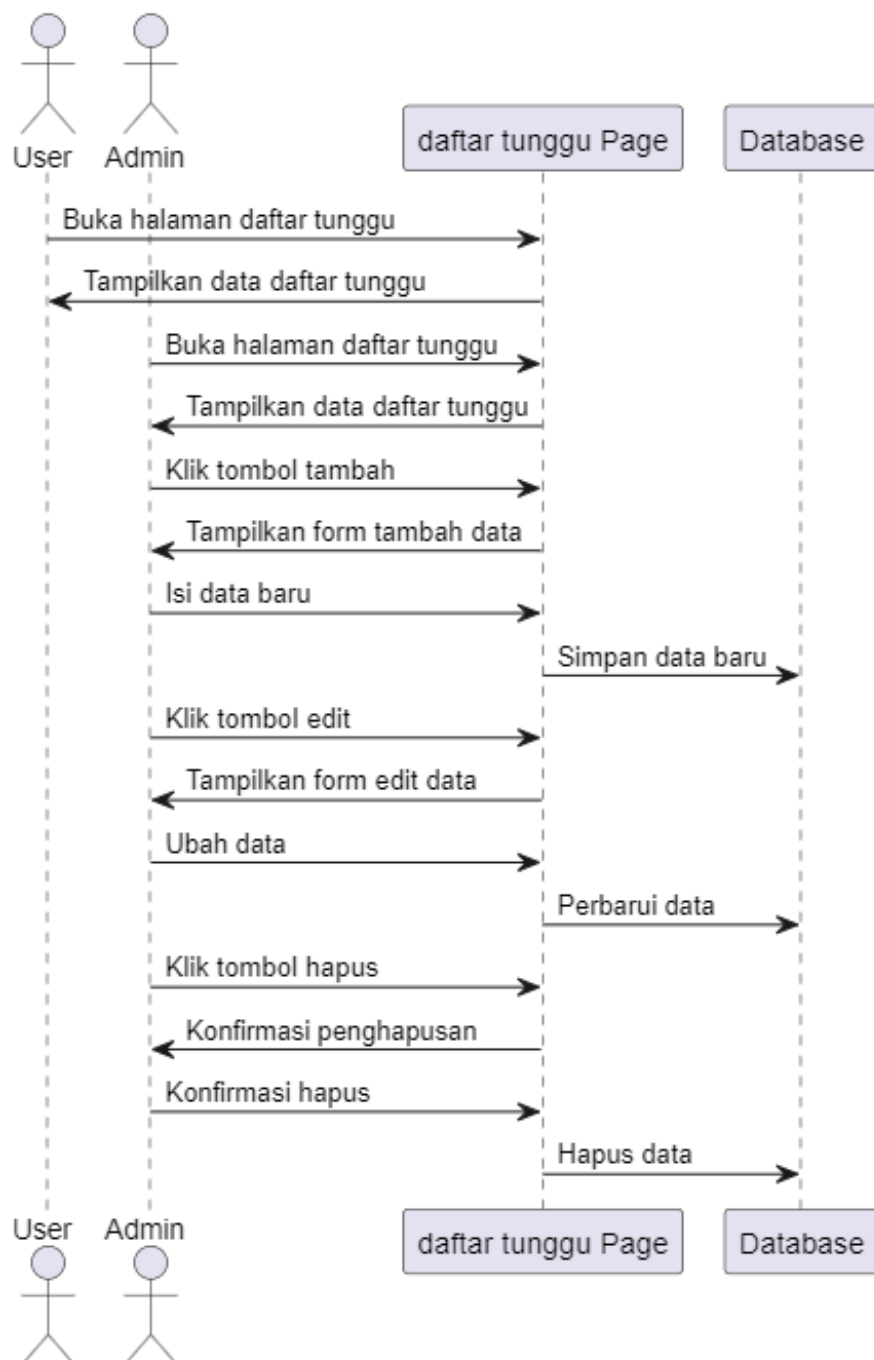


Gambar 3.5 Sequence Diagram intrap2tl

3. Sequence diagram daftar tunggu

Proses dimulai ketika Admin berhasil masuk ke *Dashboard* Aplikasi. Admin dan user dapat memilih daftar

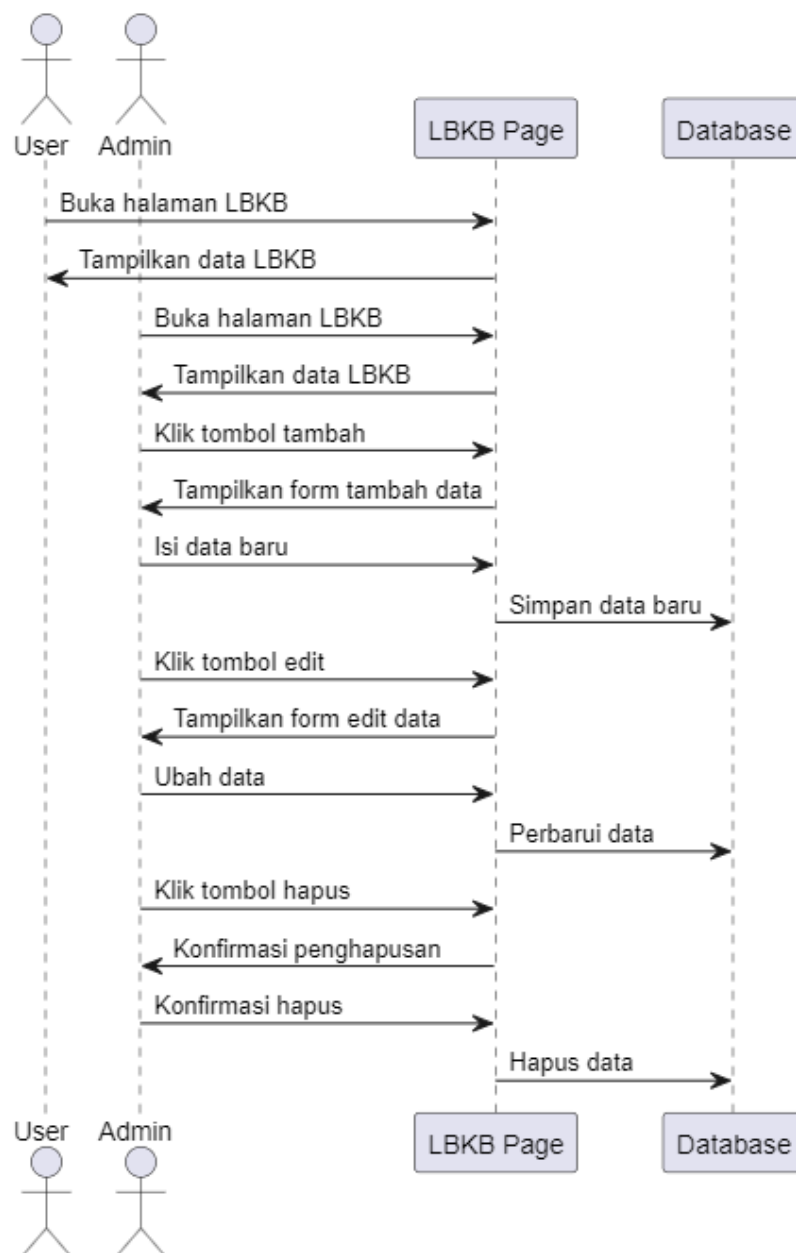
tunggu, kemudian admin dapat menambah, mengubah, menghapus, data dari menu tersebut.



Gambar 3.6 Sequence Diagram daftar tunggu

4. Sequence diagram LBKB

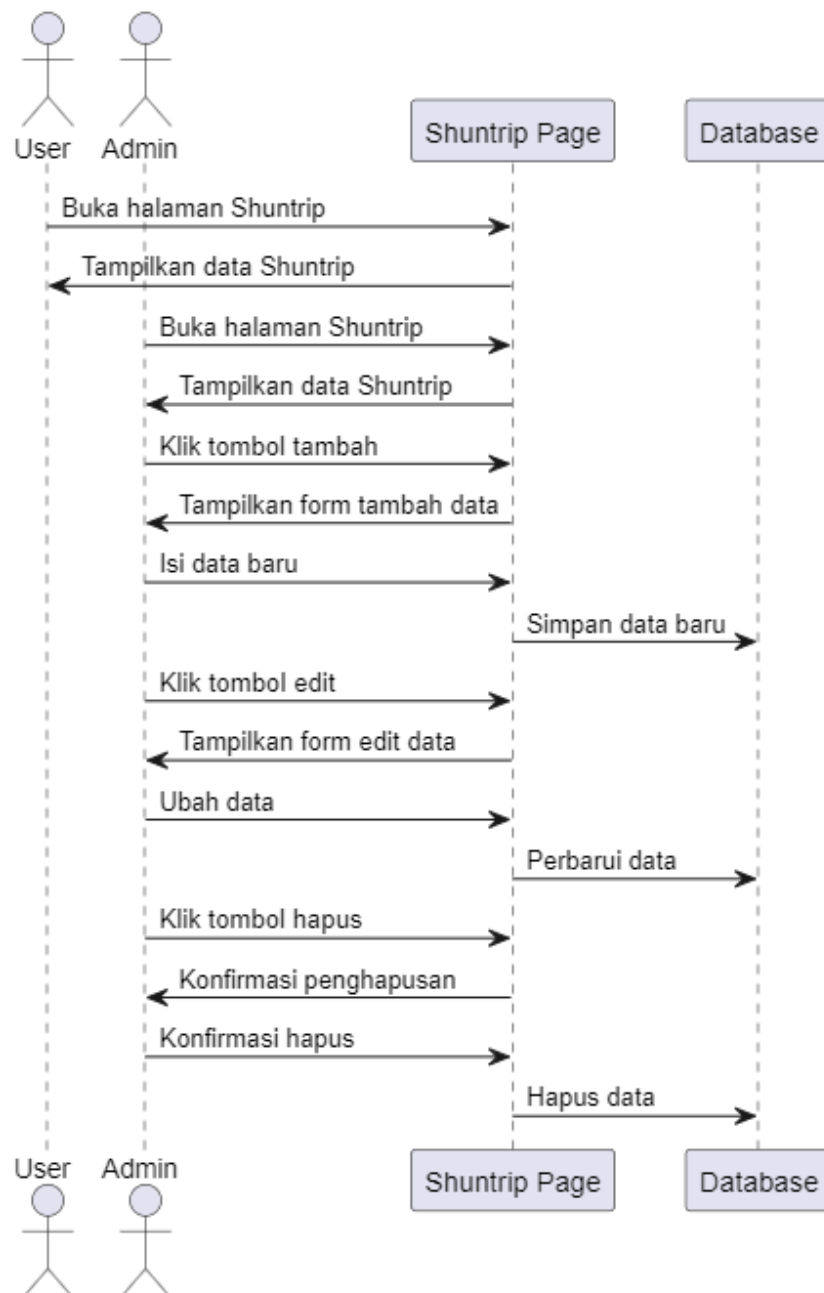
Proses dimulai ketika Admin berhasil masuk ke *Dashboard* Aplikasi. Admin dan user dapat memilih LBKB, kemudian admin dapat menambah, mengubah, menghapus, data dari menu tersebut.



Gambar 3.7 Sequence Diagram LBKB

5. Sequence diagram Shuntrip

Proses dimulai ketika Admin berhasil masuk ke *Dashboard* Aplikasi. Admin dan user dapat memilih LBKB, kemudian admin dapat menambah, mengubah, menghapus, data dari menu tersebut.



Gambar 3.8 Sequence Diagram LBKB

3.3.3.4 Class Diagram

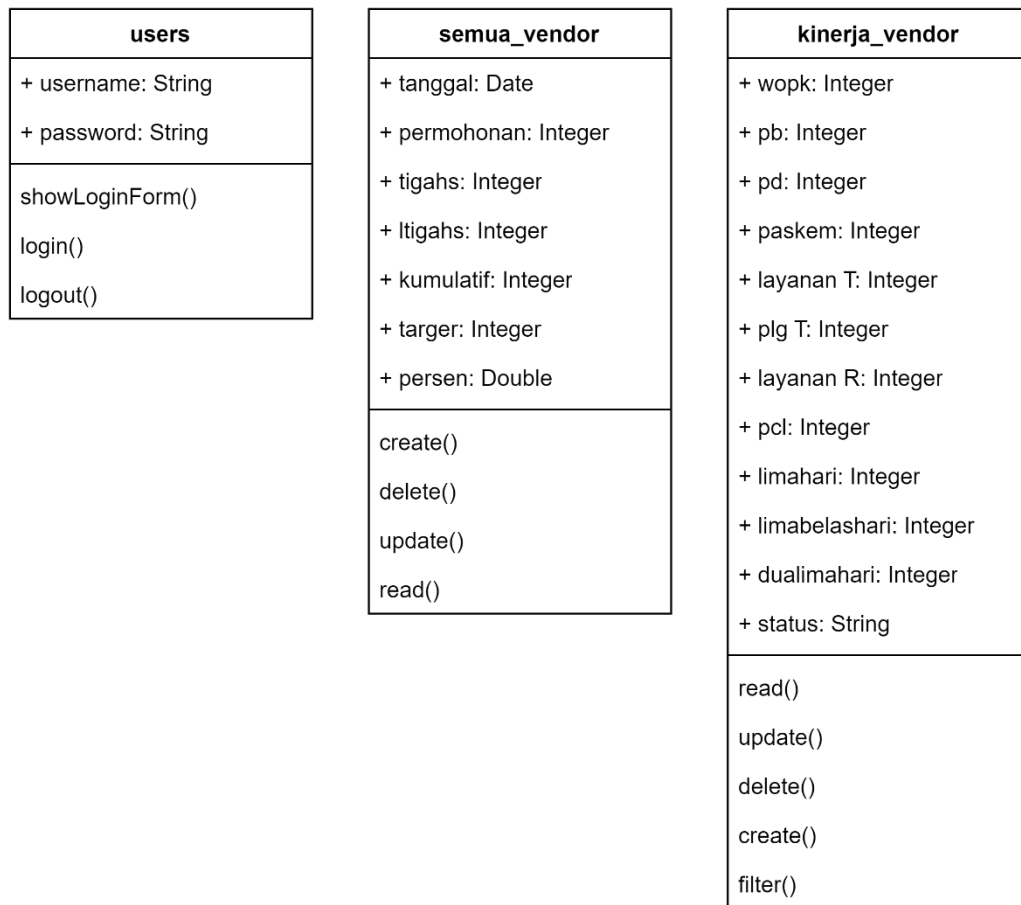
Class diagram adalah salah satu jenis diagram berbentuk struktur pada model UML (*Unified Modeling Language*). Diagram ini menggambarkan struktur, atribut, kelas, hubungan, dan metode dengan sangat jelas dari setiap objeknya. *Class* diagram memberikan data berupa hubungan apa yang terjadi di antara kelas-kelas, bukan menjelaskan kejadiannya.

Berikut adalah gambar-gambar dari *class* diagram:

1. Class diagram LBKB, Shuntrip, dan Intrap2tl

LBKB	Shuntrip	Intrap2tl
+ Tanggal: Date + Bulan: String + Tahun: Integer + Tahun: Integer + Tahun: Integer	+ Tanggal: Date + Bulan: String + Tahun: Integer + Tahun: Integer + Tahun: Integer	+ Tanggal: Date + Bulan: String + Tahun: Integer + Tahun: Integer + Tahun: Integer
create() read() update() delete() filter()	create() read() update() delete() filter()	create() read() update() delete() filter()

Gambar 3.9 tabel *class* diagram LBKB, Shuntrip, Intrap2tl



Gambar 3.10 Class Diagram users, semua_vendor, dan kinerja_vendor

3.3.3.5 Rancangan *Database*

Struktur *database* terdiri dari 6 table yang dimana menyimpan data dari tiap-tiap halaman yang ada. Nama dari tiap database ini adalah sebagai berikut:

1. shuntrip.

Tabel 3.1 Tabel shuntrip.

No	Nama <i>Field</i>	Tipe	Panjang	Keterangan
1	Id	int	11	<i>Primary key, auto_increment</i>

2	bulan	varchar	20	-
3	Target (pelanggan)	int	10	-
4	Realisasi (pelanggan)	int	10	-

2. intrap2tl (Penertiban Pemakaian Tenaga Listrik).

Tabel 3.2. intrap2tl.

No	Nama Field	Tipe	Panjang	Keterangan
1	Id	int	11	Primary key, auto_increment
2	bulan	varchar	20	-
3	target	int	10	-
4	realisasi	int	10	-

3. LBKB (Laporan Bulanan Kelainan Baca meter).

Tabel 3.3 LBKB

No	Nama Field	Tipe	Panjang	Keterangan
1	Id	int	11	<i>Primary key, auto_increment</i>
2	bulan	varchar	20	-

3	Target (pelanggan)	int	10	-
4	Realisasi (pelanggan)	int	10	-

4. users.

Tabel 3.4 Users

No	Nama Field	Tipe	Panjang	Keterangan
1	Id	int	20	Primary key, auto_increment
2	Username	varchar	255	-
3	password	int	255	-

5. Semua Vendor.

Tabel 3.5 Semua Vendor

No	Nama Field	Tipe	Panjang	Keterangan
1	Id	int	10	Primary key, auto_increment
2	tanggal	date	-	-
3	Permohonan (pelanggan)	varchar	10	-

4	Tigahs (3 jam)	int	10	-
5	Ltigahs (lebih dari 3 jam)	int	10	-
6	kumulatif	float	-	-
7	target	int	10	-
8	persen	int	10	-

6. Kinerja Vendor.

Tabel 3.6 Kinerja Vendor

No	Nama Field	Tipe	Panjang	Keterangan
1	id	int	20	Primary key, auto_increment
2	tanggal	date	-	-
3	wopk (work order perintah kerja)	int	30	-
4	pb (pasang baru)	int	30	-
5	pd (perubahan daya)	int	30	-
6	Paskem (Pasang kembali)	int	30	-
7	Layanan T (target)	varchar	50	-
8	Plg T (pelanggan target)	int	30	-

9	Layanan R (realisasi)	int	30	-
10	plc (Perubahan listrik cepat)	int	30	-
11	limahari	int	30	-
12	limabelashari	int	30	-
13	dualimahari	int	30	-
14	status	int	30	-