BAB 3
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis sistem merupakan uraian dari suatu sistem utuh yang terbagi ke dalam bagian-bagian komponen dengan maksud untuk mengidentifikasi dan mengevaluasi permasalah-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikan. Analisis dapat juga diartikan sebagai penelitian atas sistem yang telah ada dengan tujuan untuk merancang sistem yang baru atau diperbarui. Dalam proses pembuatan suatu sistem mutlak harus dilakukan penelitian dan penganalisaan tentang sistem yang akan dibangun.

3.2 Analisis Masalah

SMKN 1 Katapang mempunyai peraturan sekolah yang cukup ketat, siswa/siswi di sekolah ini tidak diijinkan untuk membawa kendaraan bermotor dan diparkirkan di area sekolah kecuali sudah mempunyai SIM (Surat Ijin Mengemudi) sehingga membuat beberapa siswa yang tidak memliki SIM tetap membawa kendaraan bermotor, siswa yang tidak mempunyai SIM tersebut memarkirkan kendaraan bermotornya di luar lingkungan sekolah, hal tersebut mengakibatkan keamanan kendaraan bermotor siswa tidak terjamin. Sehingga pihak sekolah membuat sistem parkir untuk siswa SMKN 1 Katapang agar kendaraan bermotor milik siswa terjamin keamanannya.

Sistem perpustakaan yang sedang berjalan mengalami proses sirkulasi perpustakaan yang lambat dikarenakan siswa yang datang harus menulis manual data siswa sebagai data pengunjung, proses peminjaman, pengembalian dan pengecekan waktu peminjaman masih manual dengan cara melihat kartu yang biasa ada di setiap buku perpustakaan. Maka dibutuhkan teknologi untuk membantu mempercepat proses sirkulasi di perpustakaan.

proses absensi masih dilakukan secara manual dengan cara memanggil siswa satu persatu dan menulisnya di buku absen, proses tersebut mengurangi waktu belajar. Proses rekap absen harian dilakukan oleh guru piket yang berkeliling dan melihat buku absen kelas. Hal tersebut dinilai masih kurang baik karena proses rekapitulasi yang masih manual sehingga membuat proses rekapitulasi lambat jika data yang dibutuhkan banyak. Maka dari itu dibutuhkan sistem yang terkomputerisasi sehingga dapat membantu proses rekapitulasi menjadi lebih cepat.

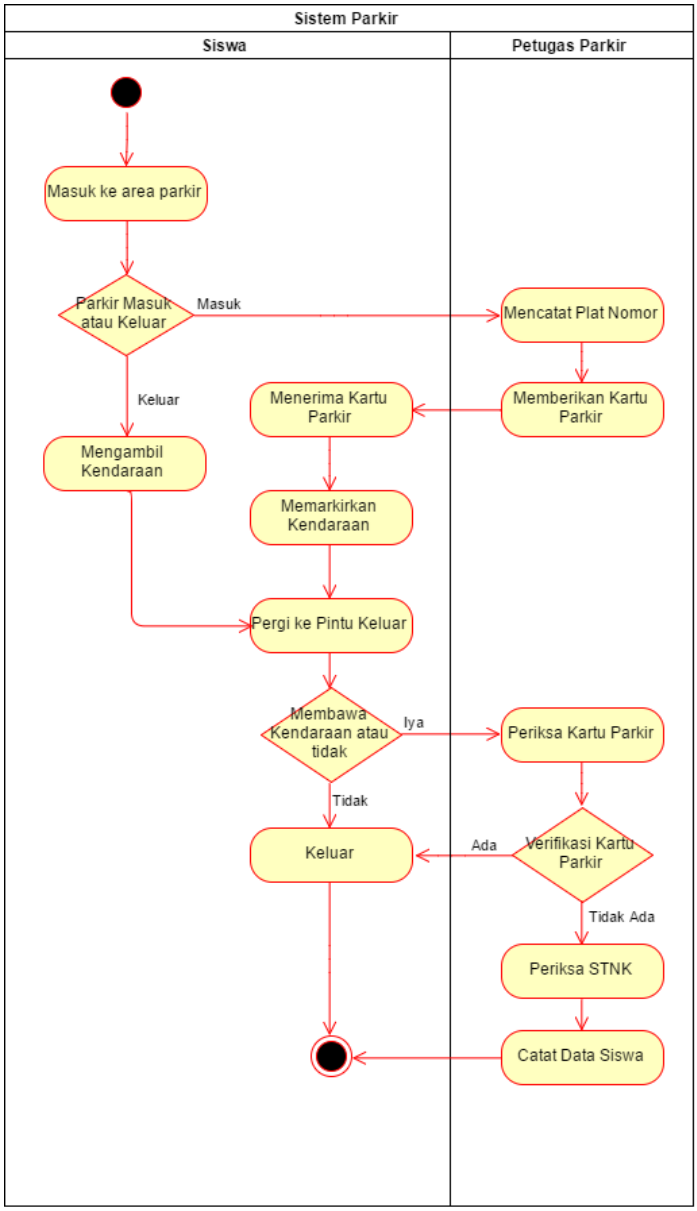
Berdasarkan analisa diatas maka terdapat permasalahan seperti panjangnya antrian ketika kegiatan belajar mengajar sekolah dan pendataan absensi elektronik maka dibangun sebuah teknologi *smart card* dengan menggunakan metode *Radio frequency identification* (RFID) yang berbentuk kartu pelajar diharapkan dapat meningkatkan kualitas layanan yang terdapat di SMKN 1 Katapang.

3.3 Analisis Prosedur Yang Sedang Berjalan

Analisis sistem berjalan merupakan proses penguraian terhadap sistem pembayaran yang telah ada atau sedang berjalan dengan tujuan untuk melakukan evaluasi yang ada, merumuskan tujuan yang ingin dicapai dengan sistem yang baru, serta menyusun rencana pembangunan sistem.

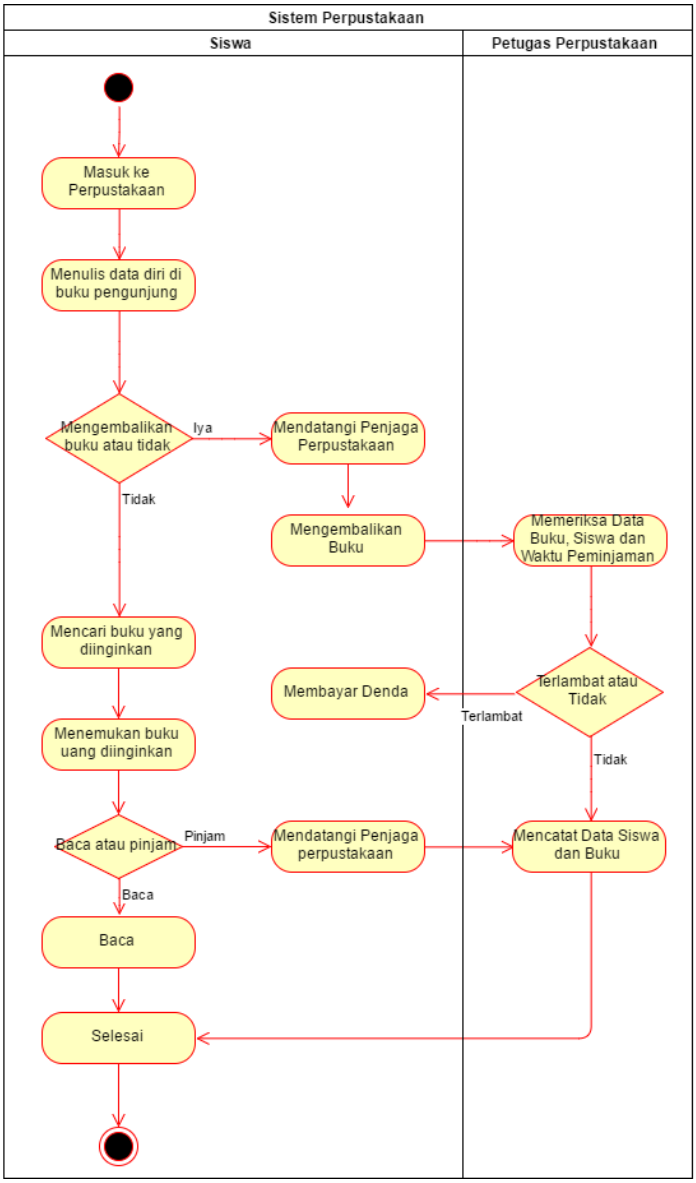
Pada dasarnya prosedur manual yang ada di SMKN 1 Katapang sama seperti proses yang ada di sekolah, proses ini terbagi menjadi 3 proses yaitu proses parkir, proses perpustakaan dan proses absensi.

Proses parkir dimulai pada saat siswa datang ke tempat parkir dan memarkirkan kendaraannya di lingkungan atau tempat parkir sekolah, kemudian petugas parkir memberikan karcis kepada siswa dan mengambil karcis dari siswa pada saat siswa keluar dari area parkir sekolah.



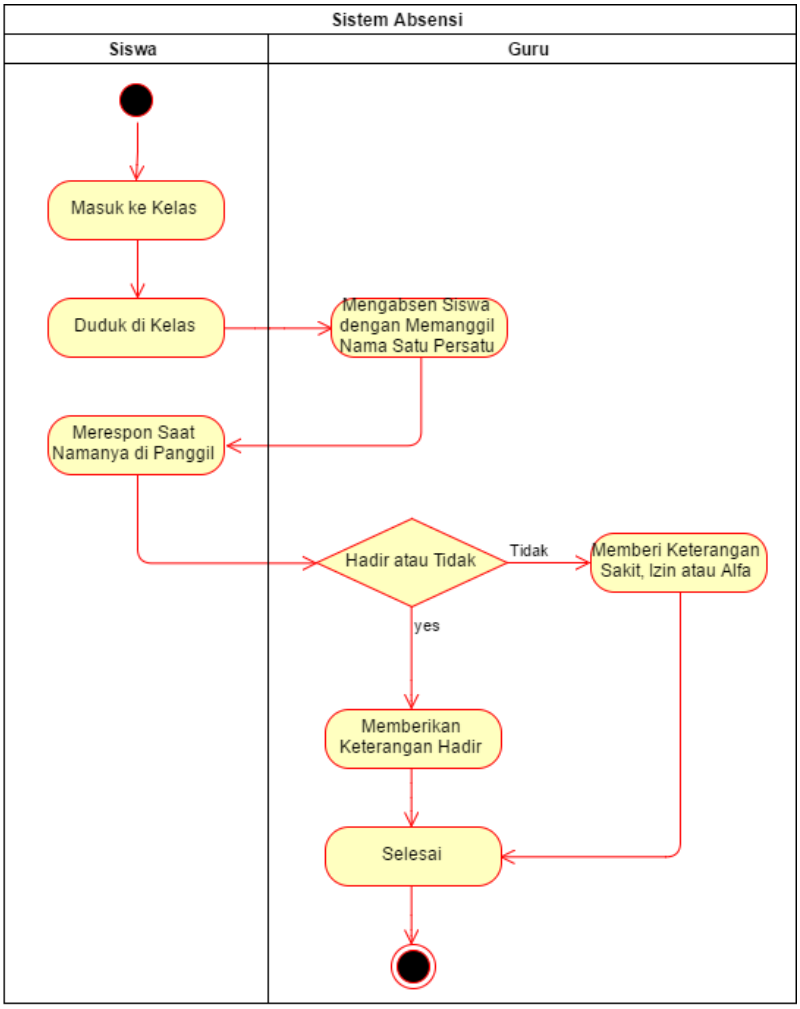
Gambar 3.1 Sistem Parkir yang Sedang Berjalan

Proses perpustakaan dimulai pada saat siswa datang ke perpustakaan kemudian mengisi buku pengunjung dengan nama dan kelas, kemudian siswa mencari buku yang ingin di pinjam atau di baca, jika ingin meminjam siswa datang ke tempat penjaga perpustakaan kemudian memberikan buku yang ingin di pinjam dan kartu perpustakaan untuk proses peminjaman, petugas perpustakaan mencatat data buku dan data siswa yang meminjam buku tersebut dan memberikan tengat waktu untuk pengembalian buku tersebut.



Gambar 3.2 Sistem Pepustakaan yang Sedang Berjalan

Proses absensi dimulai saat siswa datang kemudian masuk dan duduk di kelas, kemudian guru memanggil nama siswa satu persatu, siswa yang dipanggil namanya merespon dengan kata hadir bila hadir di dalam kelas, siswa yang tidak hadir akan mendapatkan keterangan seperti alfa, izin, atau sakit.



Gambar 3.3 Sistem Absensi yang Sedang Berjalan

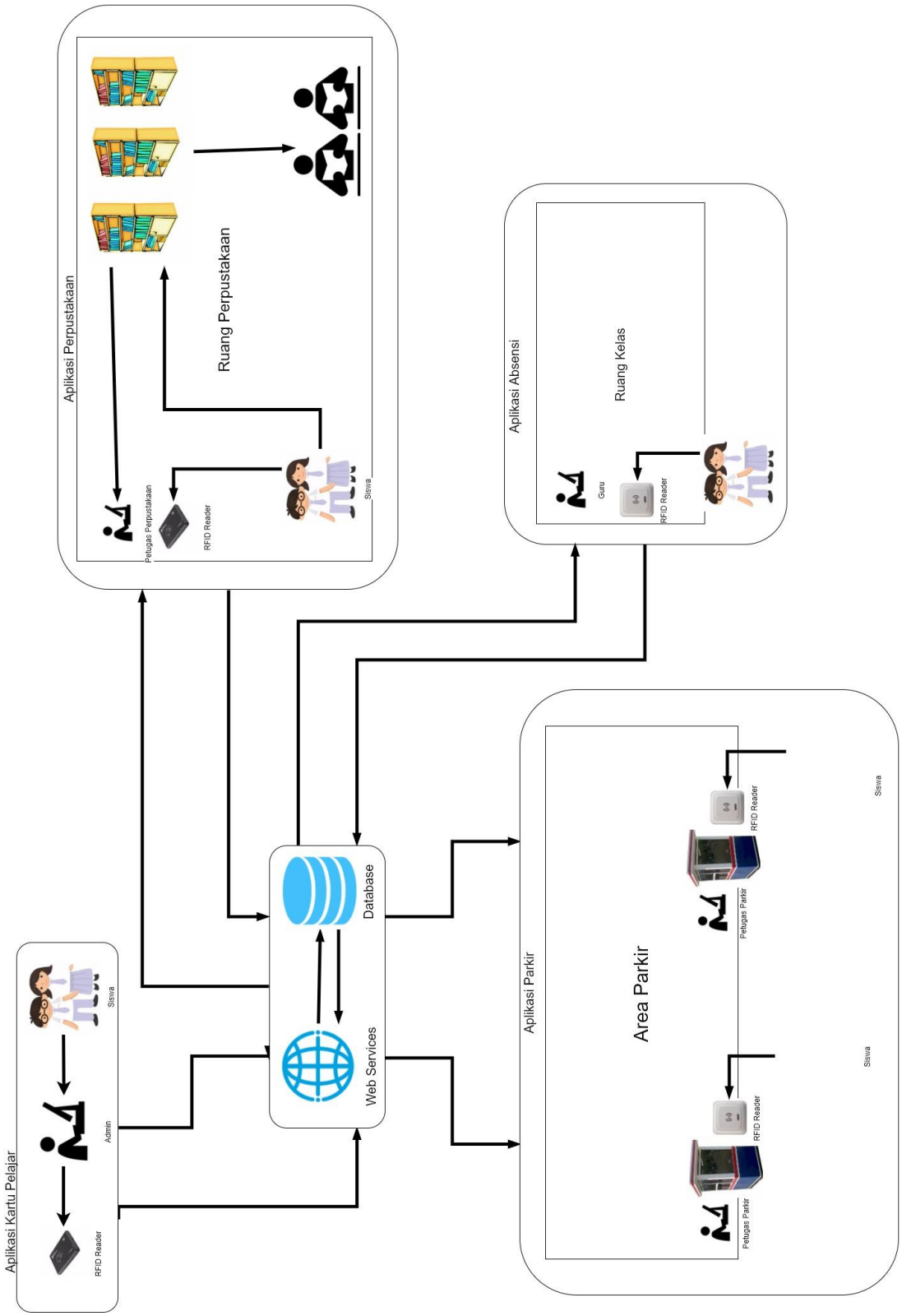
3.4 Analisis Sistem Yang Akan Dibangun

Analisis sistem yang akan dibangun merupakan penguraian dari suatu sistem yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan sebagai perbaikan-perbaikan.

3.4.1 Deskripsi Sistem

Untuk menyelesaikan masalah yang berada di SMKN 1 Katapang, diperlukan aplikasi yang terdiri dari server yang juga berfungsi sebagai sistem yang menyimpan *history* data transaksi yang dilakukan oleh kartu pelajar pintar

yang dimiliki oleh siswa. Selain sistem server dibutuhkan juga sistem untuk *write* dan *read* yang akan digunakan untuk keperluan parkir, perpustakaan dan absensi. Sistem *write* dan *read* digunakan oleh petugas parkir, perpustakaan, dan absensi. Petugas parkir akan membaca data siswa dan melakukan proses transaksi jika diperlukan, petugas perpustakaan akan membaca data siswa dan melakukan *input* data buku ke dalam database *center*, absensi akan melakukan proses kehadiran siswa ke dalam database *center*.



Gambar 3.4 Sistem yang Akan dibangun

1. Aplikasi Kartu Pelajar

Pendaftaran kartu pelajar berteknologi RFID ini dilakukan oleh admin dengan menggunakan data yang sudah dimiliki sekolah, adapun siswa harus menambahkan data nomor polisi kendaraan bermotor miliknya dan kepemilikan SIM, jika ingin membawa kendaraan dan menggunakan fasilitas parkir SMKN 1 Katapang. Admin dapat mengelola data siswa, data guru, data kelas jika ada murid baru, murid pindah, guru baru, guru pindah dan penambahan atau pengurangan kelas.

2. Aplikasi Parkir

Siswa SMKN 1 KATAPANG yang telah memiliki kartu pelajar berbasis RFID dan ingin memarkirkan kendaraannya wajib menempelkan kartu pelajar pada RFID reader yang telah terpasang di gerbang utama. Sebelumnya siswa yang memiliki kendaraan bermotor wajib mendaftarkan nomor polisi kendaraan bermotornya. Sistem kerja kartu pelajar RFID ini adalah dengan menempelkan kartu pada RFID reader, setelah kartu pelajar tersebut terbaca identitasnya oleh RFID reader maka portal otomatis terbuka. Dengan terbukanya portal maka RFID siswa tersebut telah terdaftar sesuai dengan database.

3. Aplikasi Perpustakaan

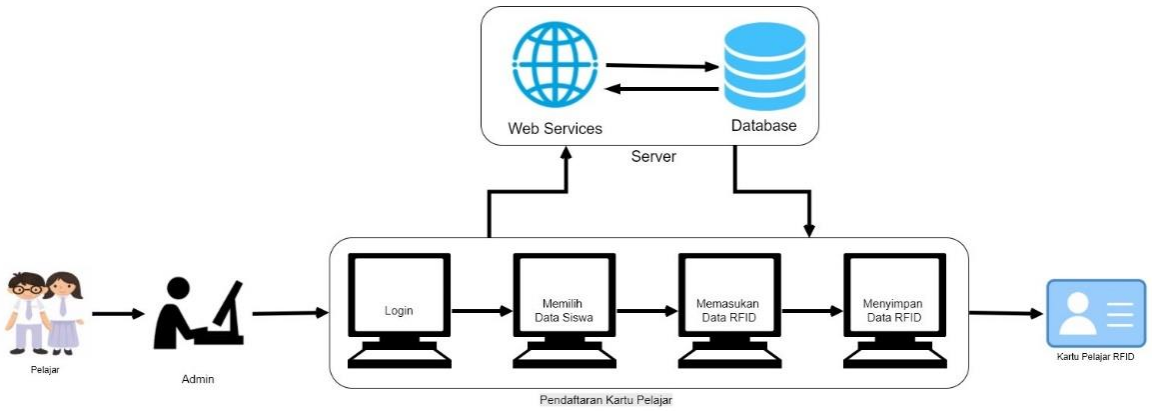
Untuk mempermudah dan mengefisiensikan waktu, kartu pelajar RFID ini dapat digunakan pada saat siswa mengunjungi perpustakaan. Dengan cara menempelkan kartu pelajar pada RFID reader yang berada di meja petugas perpustakaan, jika siswa ingin meminjam buku siswa dapat memilih buku yang akan dipinjam dan membawanya ke petugas perpustakaan. Dengan digunakannya Kartu Pelajar RFID ini petugas perpustakaan tidak perlu lagi mencatat data siswa, petugas perpustakaan hanya memasukan kode buku dan lama peminjaman buku karena data yang sudah ada di dalam database, ketika siswa mengembalikan buku ke perpustakaan petugas perpustakaan cukup menempelkan RFID milik siswa ke RFID *reader* untuk mendapatkan informasi siswa dan buku yang dipinjam,

komputer akan mengkonfirmasi keterlambatan buku yang dipinjam dan denda bila pengembalian buku tersebut terlambat. Jika ada koleksi buku perpustakaan baru petugas perpustakaan dapat langsung memasukan info buku ke dalam database.

4. Aplikasi Absensi

Kartu pelajar berteknologi RFID ini digunakan juga untuk melakukan absensi kelas. Sistem absensi dipergunakan dengan cara menempelkan kartu pada RFID reader, kemudian program akan membaca identitas siswa, aplikasi menyatakan siswa hadir pada absensi sekolah dan mencatat waktu ketika siswa melakukan absen menggunakan kartu pelajar tersebut. Siswa juga wajib melakukan absen ketika hendak pulang. Ketika ada siswa yang tidak hadir guru akan mengisi keterangan kenapa siswa yang bersangkutan tidak hadir dengan keterangan sakit, izin atau tanpa keterangan. Guru pun dapat melakukan rekapitulasi absensi menggunakan aplikasi ini.

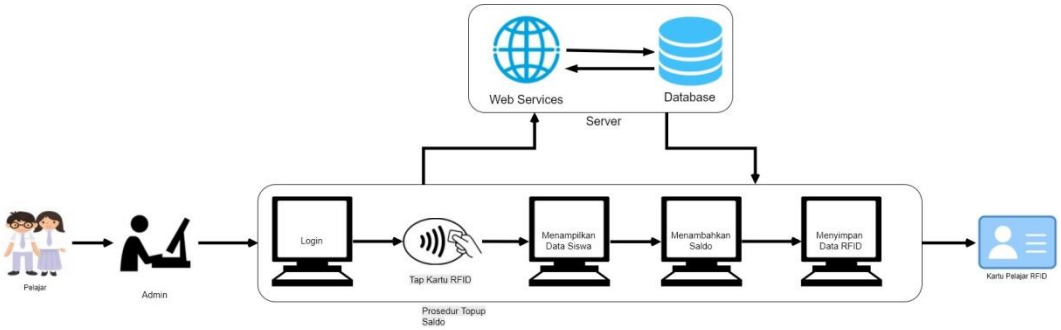
Aplikasi Kartu Pelajar



Gambar 3.5 Prosedur Pembuatan Kartu

Siswa yang belum melengkapi data nomor polisi dan kepemilikan SIM dapat memberikan informasi tersebut kepada admin, sehingga admin dapat melakukan pembaharuan terhadap data siswa tersebut, bila kartu pelajar siswa berteknologi RFID hilang atau rusak siswa dapat membuat kartu pelajar baru dengan cara datang kepada admin dan mengkonfirmasi kehilangan atau kerusakan kartu pelajar tersebut, admin akan membuat kartu pelajar baru dan memperbaharui data

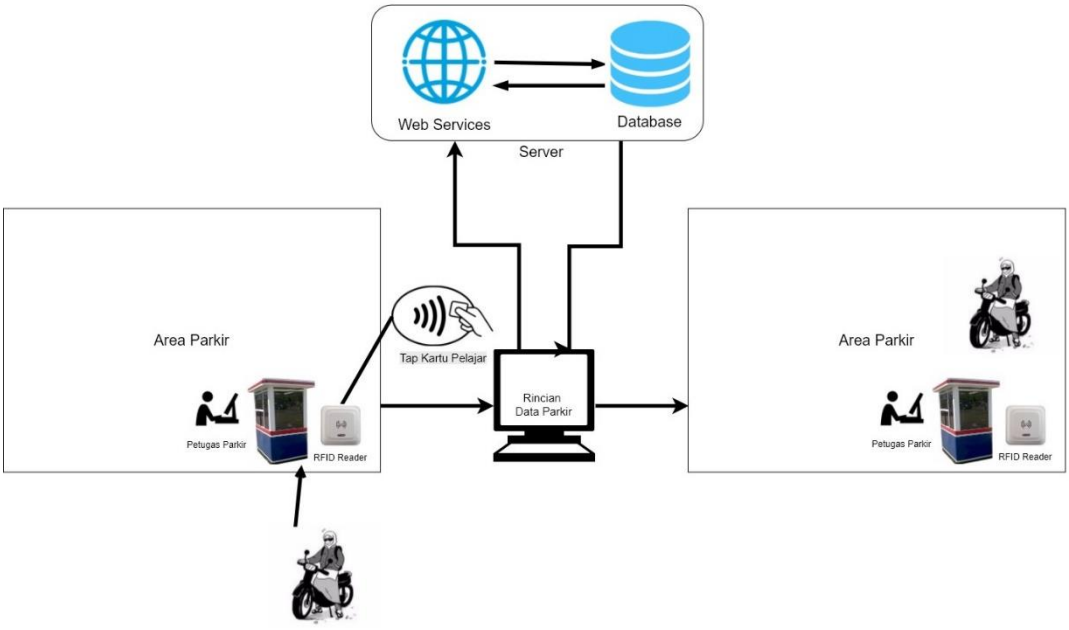
siswa dengan ID RFID baru. Admin dapat menambahkan atau menghapus data siswa, guru, dan kelas jika ada siswa atau guru yang pindah, lulus atau baru dan kelas tambahan atau pengurangan kelas.



Gambar 3.5 Prosedur Top Up Saldo

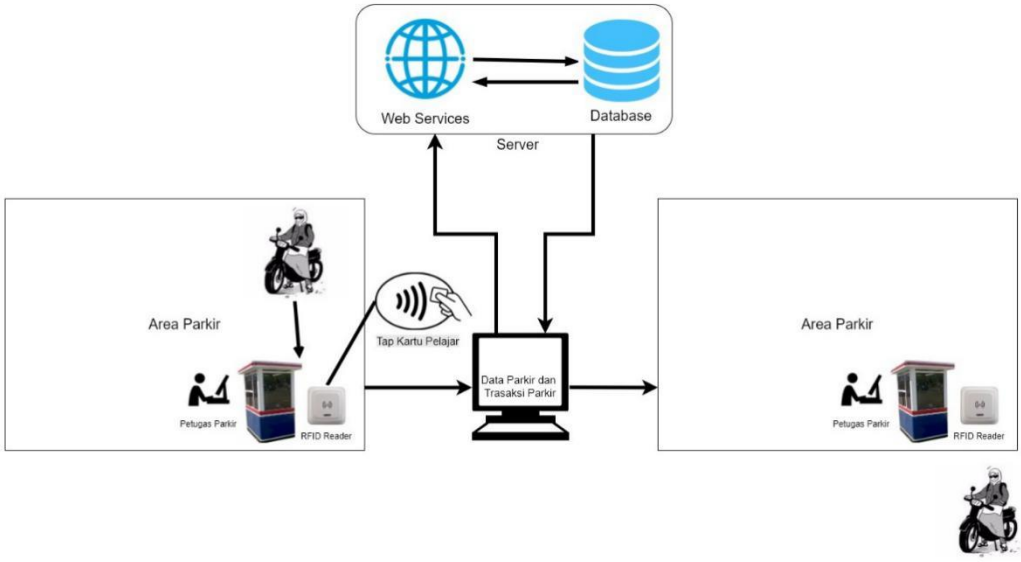
Siswa yang ingin melakukan top up saldo, datang ke admin kemudian admin akan menyentuh kartu pelajar berteknologi RFID ke RFID *reader*, setelah data siswa muncul siswa memberikan uang sebesar nominal saldo yang akan diisi, kemudian admin akan menambah saldo sesuai nominal uang yang diberikan oleh siswa.

Aplikasi Parkir



Gambar 3.7 Prosedur Parkir Masuk

Prosedur parkir masuk untuk siswa SMKN 1 Katapang bermula dengan siswa yang membawa kendaraan sebagai aktor memindai kartu pelajar yang telah dilengkapi oleh teknologi RFID ke RFID Reader di pos parkir masuk, sistem akan membaca ID RFID kemudian sistem akan mendeteksi apakah ID RFID tersebut terdaftar sebagai siswa SMKN 1 Katapang. Bila RFID tersebut telah terdaftar maka data parkir masuk milik siswa tersebut akan tercatat dan siswa tersebut dapat memakirkan kendaraannya di area parkir yang telah disediakan oleh pihak SMKN 1 Katapang, akan tetapi jika RFID tersebut tidak terdaftar akan muncul pesan bahwa RFID tidak terdaftar dan siswa tidak dapat memarkirkan kendaraannya.

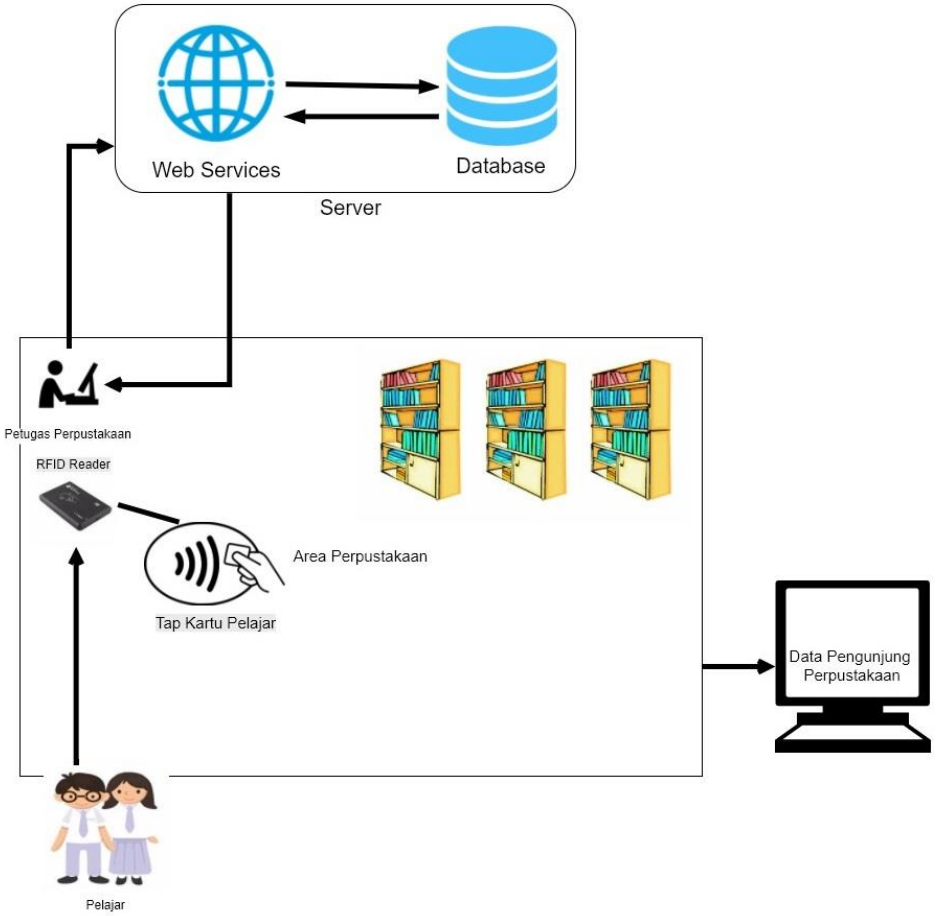


Gambar 3.8 Prosedur Parkir Keluar

Siswa yang telah memiliki kartu pelajar berteknologi RFID mengambil kendaraan miliknya yang telah diparkir di area parkir SMKN 1 katapang. Ketika siswa pulang, siswa menempelkan kembali kartu pelajar berteknologi RFID ke RFID *reader*, sistem akan membaca ID RFID dan mengkonfirmasi data siswa, data kepemilikan SIM dan jam masuk parkir, setelah sistem telah mengkonfirmasi data tersebut, petugas parkir harus melakukan pengecekan apakah nomor polisi kendaraan bermotor milik siswa tersebut sama dengan yang di daftarkan, bila semua prosedur berjalan dengan baik siswa dapat pulang membawa kendaraan

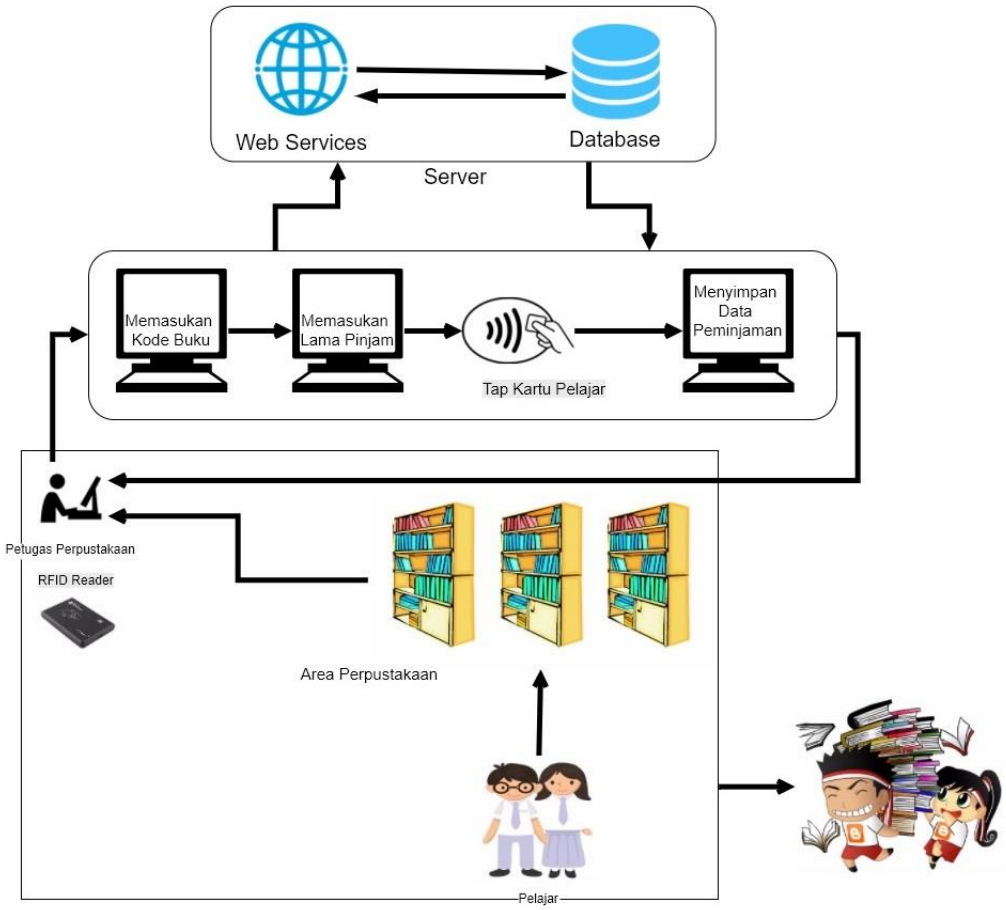
bermotor miliknya pulang dan sistem akan mencatat waktu siswa keluar dari fasilitas parkir SMKN 1 KATAPANG.

1. **Aplikasi Perpustakaan**



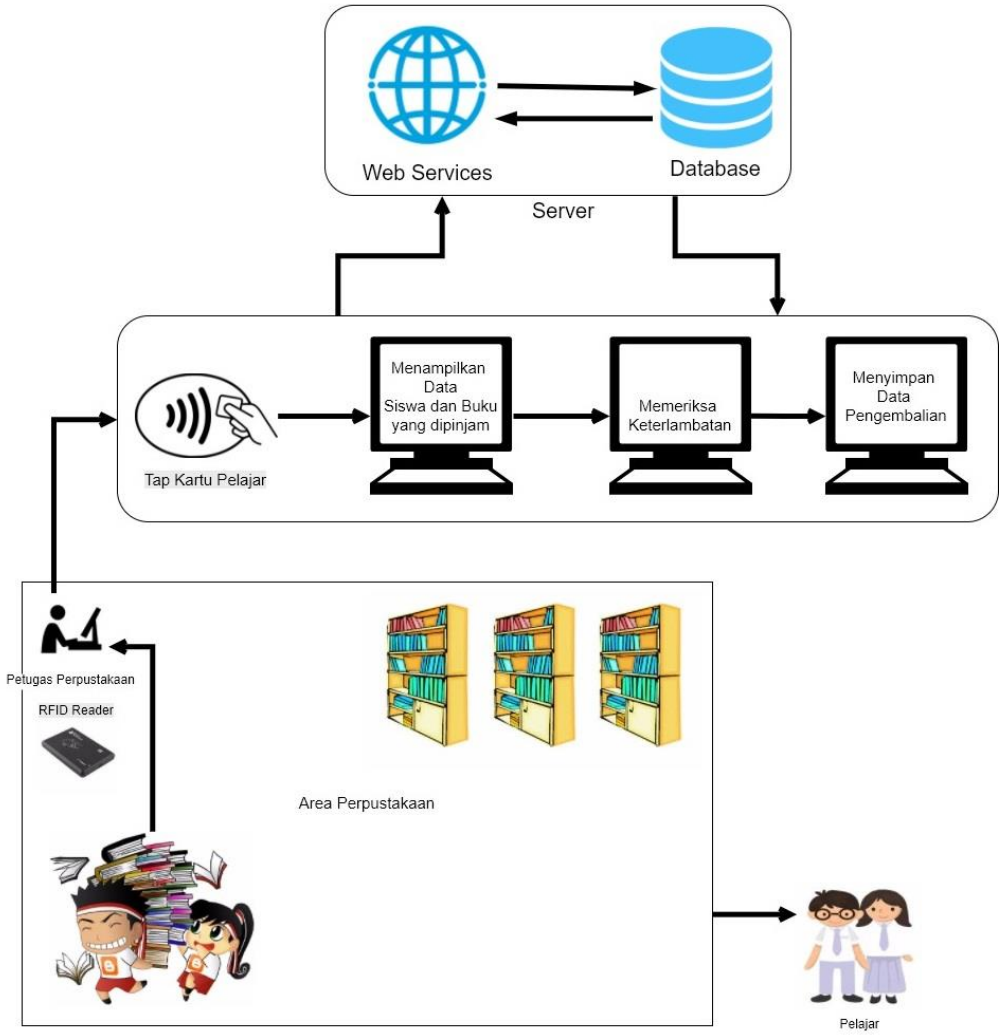
Gambar 3.9 Prosedur Pengunjung Perpustakaan

Siswa yang telah memiliki kartu pelajar berteknologi RFID dapat menggunakan fasilitas perpustakaan, ketika siswa datang berkunjung ke perpustakaan siswa harus menempelkan kartu pelajar berteknologi RFID ke RFID *reader*, kemudian sistem akan membaca ID RFID dan mencari data siswa, sistem pun mencatat waktu ketika siswa itu berkunjung ke perpustakaan. Siswa dapat mencari buku yang diinginkannya, setelah siswa menemukan buku yang diinginkanya siswa dapat membaca buku tersebut di area perpustakaan atau meminjam buku tersebut untuk di bawa pulang.



Gambar 3.10 Prosedur Peminjaman Buku Perpustakaan

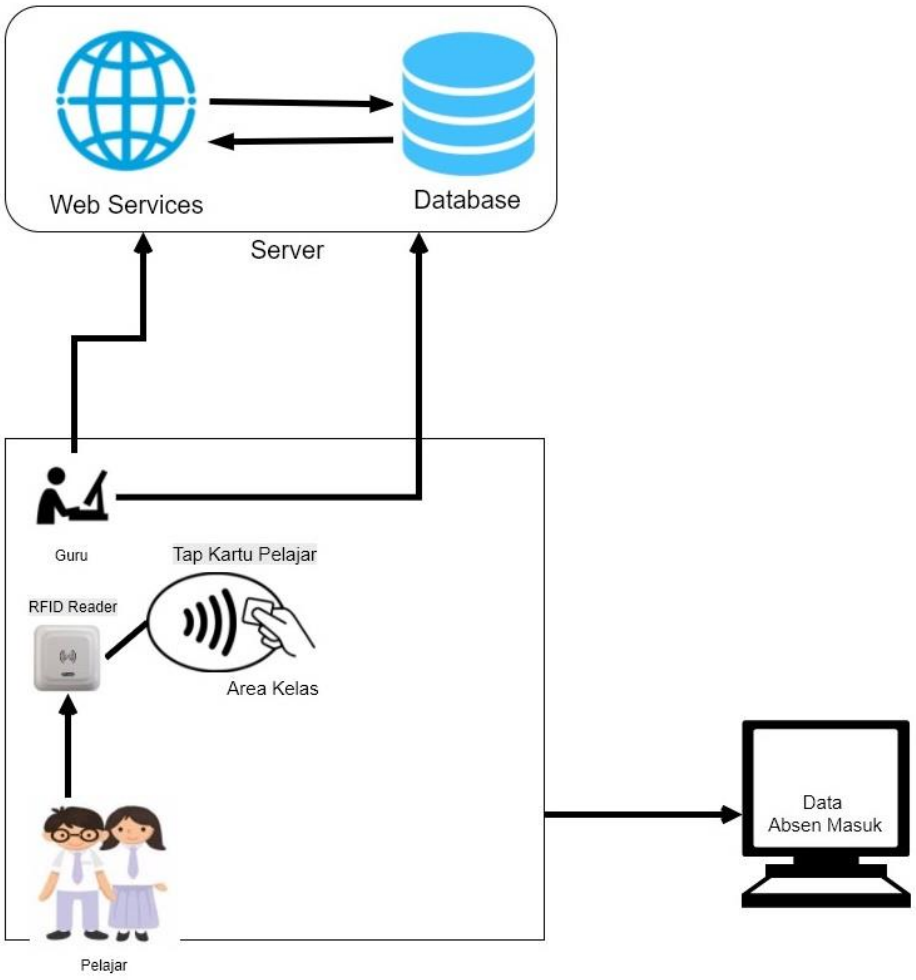
Siswa yang akan melakukan peminjaman buku, membawa buku yang ingin dipinjam ke petugas perpustakaan. Petugas perpustakaan akan melakukan input kode buku dan berapa lama peminjamannya, setelah kode buku dan lama peminjaman di masukan, petugas akan meminta siswa untuk menempelkan kartu pelajar berteknologi RFID ke RFID *reader*, ketika RFID *reader* membaca ID RFID sistem akan mencari data siswa dan menambahkan data peminjaman buku.



Gambar 3.11 Prosedur Pengembalian Buku Perpustakaan

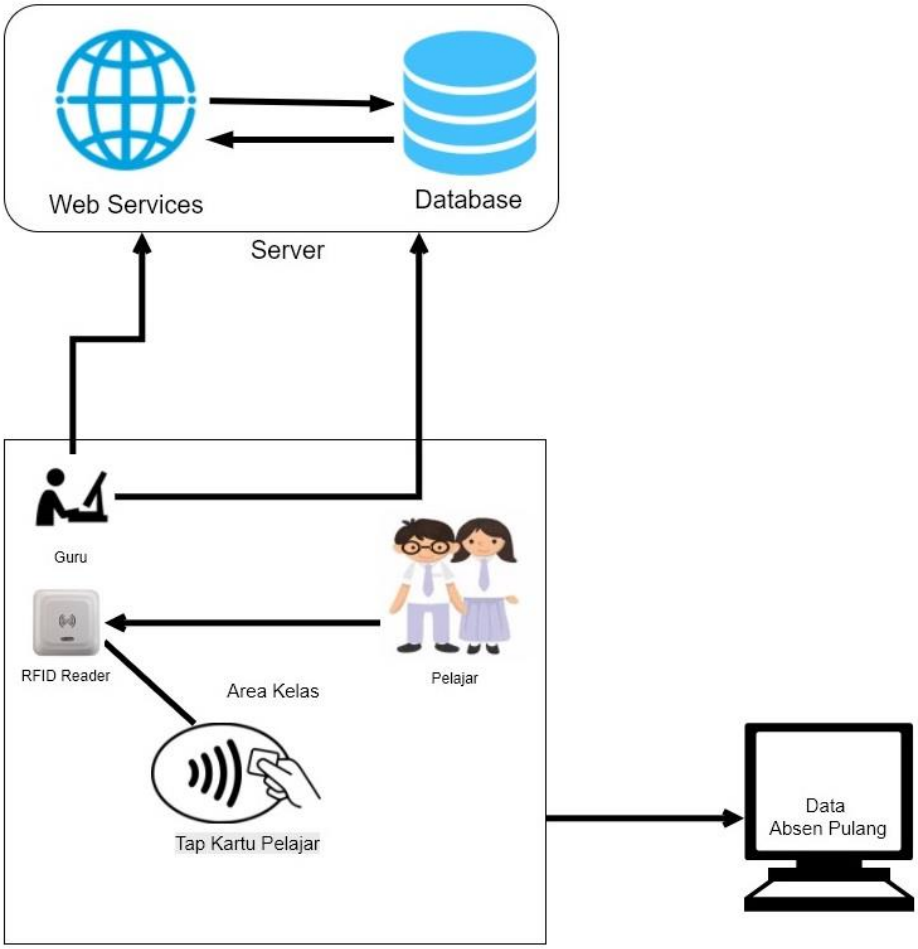
Bila siswa ingin mengembalikan buku yang dipinjam, siswa cukup datang ke petugas perpustakaan, kemudian memberikan buku tersebut kepada petugas perpustakaan, petugas perpustakaan akan memintas siswa untuk menempelkan kartu pelajar berteknologi RFID ke RFID *reader*, sistem akan mencari data siswa dan data peminjaman buku, kemudian sistem akan menghitung waktu keterlambatan buku tersebut.

2. Aplikasi Absensi



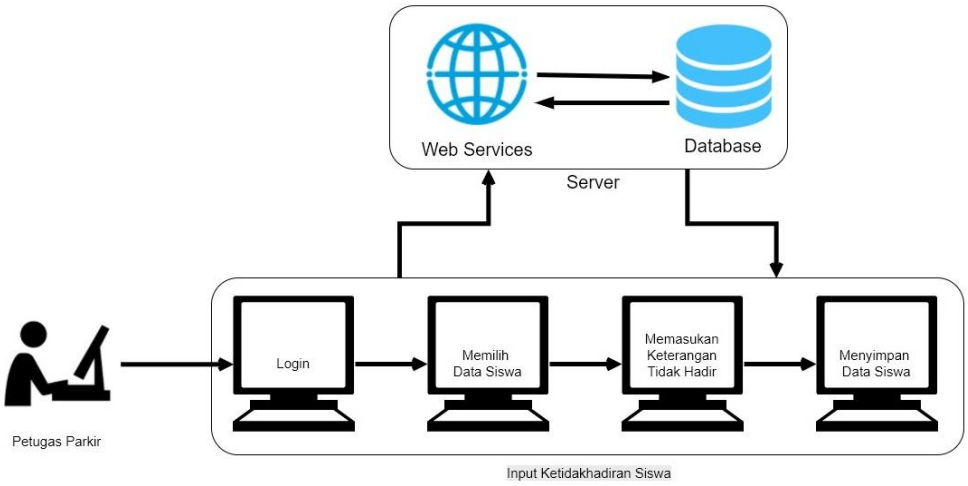
Gambar 3.6 Prosedur Absensi Datang

Sebelum siswa memulai kegiatan belajar, siswa harus menempelkan kartu pelajar berteknologi RFID ke RFID *reader* untuk melakukan proses absensi, kemudian sistem akan mencari data siswa, setelah data siswa divalidasi, sistem akan mencatat waktu kedatangan siswa dan memberi kehadiran pada absensi hari tersebut.



Gambar 3.7 Prosedur Absensi Pulang

Setelah kegiatan belajar selesai siswa harus menempelkan kartu pelajar berteknologi RFID ke RFID *reader* untuk melakukan proses absensi kembali, sistem akan mencari data siswa dan memvalidasi data, kemudian mencatat waktu siswa pulang, dan proses absensi hari tersebut selesai.

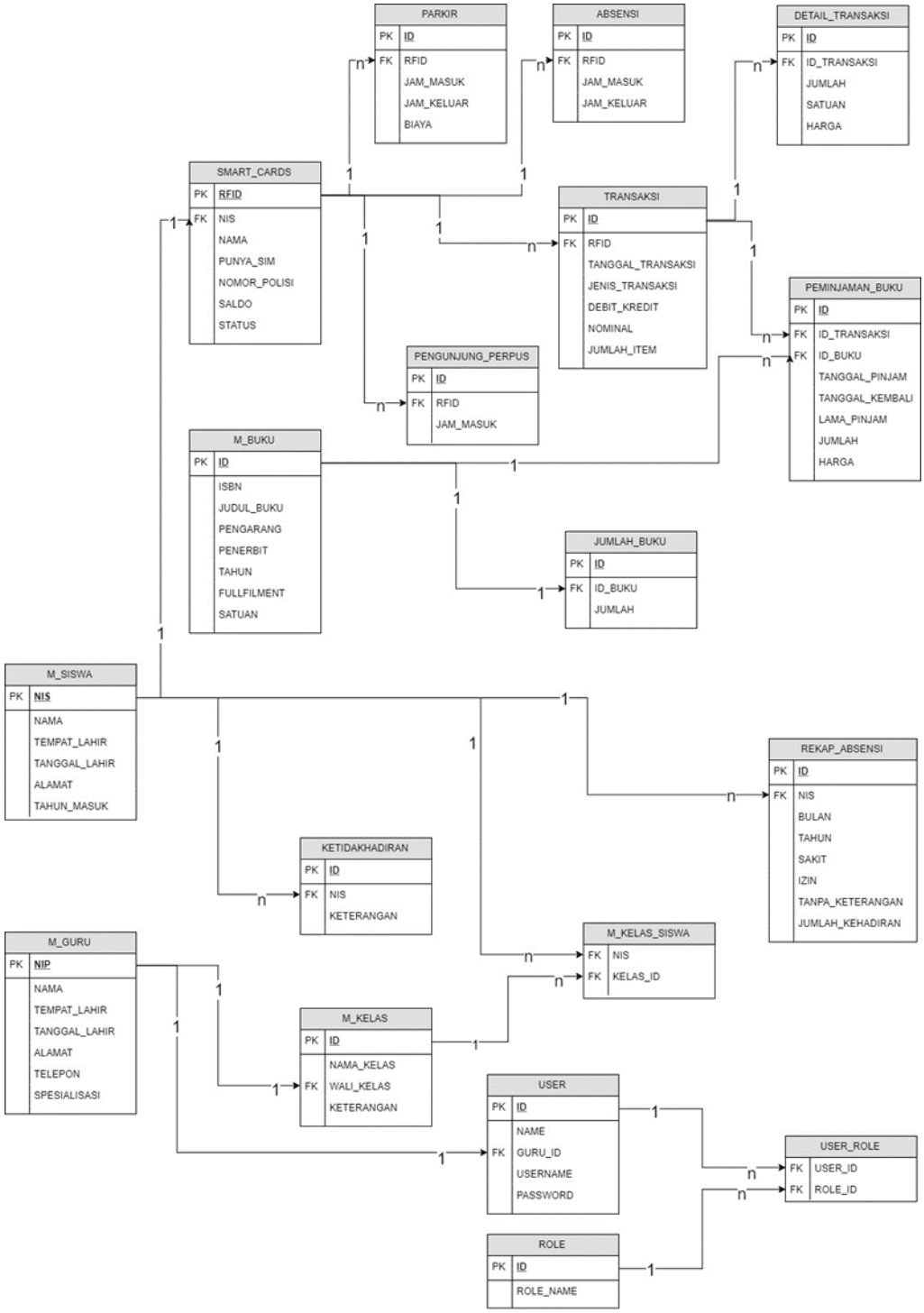


Gambar 3.8 Prosedur Mengisi Keterangan Tidak Hadir

Jika ada siswa yang tidak melakukan absensi menggunakan kartu pelajar berteknologi RFID, guru dapat memberikan keterangan tidak hadir seperti sakit, izin atau tanpa keterangan.

3.5 Analisis Data

Analisis ini bertujuan untuk mempermudah dalam perancangan informasi, agar data-data tersebut dapat menyajikan informasi yang benar maka data-data tersebut harus saling berhubungan atau berelasi antara satu dengan yang lainnya. Adapun hubungan antar data-data tersebut akan dimodelkan ke dalam skema relasi, gambar skema relasi dapat di lihat pada gambar 3.14:



Gambar 3.9 Skema Relasi

3.5.1 Analisis Data Parkir

Siswa yang memarkirkan kendaraannya di fasilitas parkir SMKN 1 KATAPANG sebelumnya hanya di berikan tiket parkir yang di buat sendiri oleh pihak sekolah, contoh kartu parkir dapat dilihat pada gambar 3.15.



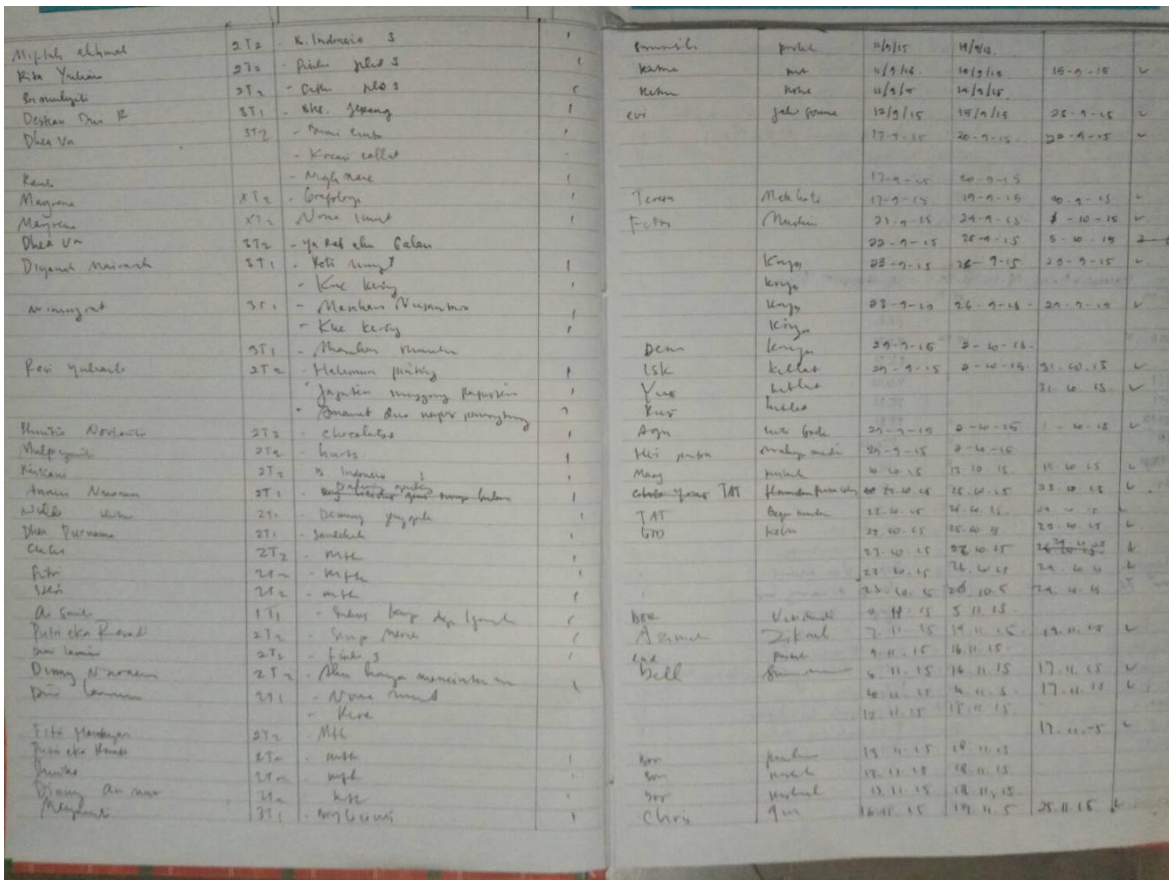
Gambar 3.10 Kartu Parkir

Petugas parkir hanya mengkonfirmasi menggunakan nomor parkir yang tertera di tiket untuk meyakinkan siswa tersebut adalah pemilik kendaraan bermotor tersebut.

3.5.2 Analisis Data Perpustakaan

Data perpustakaan untuk data pengunjung, peminjaman dan pengembalian buku masih menggunakan cara konvesional yaitu dengan menulis data pengunjung di buku dan peminjaman, pengembalian dan koleksi buku

perpustakaan menggunakan microsoft excel, contoh data peminjaman buku dapat dilihat pada gambar 3.16.



Gambar 3.11 Data Buku Perpustakaan

3.5.3 Analisis Data Absensi

Data absensi yang ada di dapat dengan menggunakan buku absen dan memanggil satu persatu nama siswa, membuat proses rekapitulasi absen harus di lakukan secara manual, contoh rekapitulasi dapat dilihat pada gambar 3.17.

Kelas : X T. Elektronika Industri 1
Wali Kelas : Dahlan Toyib,S.Pd

No.	NIS	Nama Lengkap	Bulan														Jumlah		
			Agustus			September			Oktober			November			Desember					
			S	I	A	S	I	A	S	I	A	S	I	A	S	I	A	S	I	A
1	1617.10.001	ADAM MAULDY SAEFFULLOH						1	1									1		1
2	1617.10.002	AHMAD HABIBI			3	1		8	1		6							2		17
3	1617.10.003	ALDI ANUGRAH FIRDAUS				1												1		
4	1617.10.004	ANDI AWALUDIN																		
5	1617.10.005	ANISA AZHAR																		
6	1617.10.006	ARDYAN AULIA	2															2		
7	1617.10.007	AZIS SUSANTO																		
8	1617.10.008	CAHYADI																		
9	1617.10.009	DAFFA FIRSTY FADILLAH																		
10	1617.10.010	DENY WYLY SETYADY S.P				1					1							1		1
11	1617.10.011	DINI RIZKIANI				1			2									3		
12	1617.10.012	FALDI ADAM WIBIKSANA		1															1	
13	1617.10.013	FIKRI FEBRIANSYAH																		
14	1617.10.014	FITRI AFRIANI				1					1							1		1
15	1617.10.015	GIVALDY			9	Keluar September 2016														9
16	1617.10.016	HILMAN N HAQIQI																		
17	1617.10.017	IQBAL MOCHAMMAD FAUZAN	1				1					1						2	1	
18	1617.10.018	IRFAN MIFTAHUDIN					1	1											1	1
19	1617.10.019	KHOPIPAH FIKORIYAH					2		1		1	1						2	2	1
20	1617.10.020	KIKI RINALDI			7			1			7									15
21	1617.10.021	MELLY SA WIDIARTI							1									1		
22	1617.10.022	MIKI NUR RAHMAN				1												1		
23	1617.10.023	MUHAMAD DAFFA FEBRIANA								2									2	
24	1617.10.024	MUHAMMAD AFRIZAL HAFIZ							4									4		
25	1617.10.025	MUHAMMAD EGIANSYAH QOMARA	1									1						2		
26	1617.10.026	MUHAMMAD MIFTAH FADAL		1					1		2							1	1	2
27	1617.10.027	NUR SALIM																		
28	1617.10.028	RAIHAN ZUL FIQIH				2			5			2						9		
29	1617.10.029	Rd. WILMAN GUMILAR																		
30	1617.10.030	RIFQI NAUFAL HIBATULLAH S.																		
31	1617.10.031	RISKA ROSLIANA	1						1									2		
32	1617.10.032	RIZFAN NUR ARIFIN			1	1			Keluar Oktober 2016									1		1
33	1617.10.033	RIZQY FATHURROHMAN			1															1
34	1617.10.034	SITI LUTHFIAH RAHMAH A.	1			1								1				2		1

Gambar 3.12 Data Rekapitulasi

3.6 Analisis Kebutuhan Non Fungsional

Analisis dan kebutuhan non-fungsional meliputi kebutuhan perangkat keras, kebutuhan perangkat lunak dan pengguna sistem yang akan menggunakan aplikasi. Tujuan dari analisis kebutuhan non-fungsional ini agar aplikasi yang dibangun dapat digunakan sesuai dengan kebutuhan pengguna dalam mencari informasi yang dibutuhkan.

3.6.1 Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak merupakan penguraian kebutuhan-kebutuhan non fungsional yang berhubungan dengan spesifikasi perangkat lunak dan berhubungan dengan proses pembangunan perangkat lunak.

Kebutuhan perangkat lunak:

1. Sistem Operasi yang digunakan adalah Windows 10 64-bit
2. Visual Studio 2013 64-bit
3. Notepad ++

3.6.2 Analisis Kebutuhan Perangkat Keras

Analisis kebutuhan perangkat lunak merupakan penguraian kebutuhan-kebutuhan non fungsional yang berhubungan dengan spesifikasi perangkat keras dan berhubungan dengan proses pembangunan perangkat lunak.

1. Kebutuhan Perangkat keras untuk aplikasi server :
   a) Processor yang direkomendasikan Intel Core i3 M330 2.13 GHz.
   b) RAM 2 GB atau lebih
   c) Hardisk yang direkomendasikan minimal 500 GB.
2. Kebutuhan Perangkat Keras untuk aplikasi client :
   a. Processor yang direkomendasikan AMD A8-6410 APU 2.00 Ghz
   b. RAM 2GB atau lebih
   c. Hardisk minimal 20GB (untuk Windows 7) disesuaikan dengan sistem operasi yang digunakan
3. Kebutuhan Perangkat Keras lainnya :
   a. RFID Reader 125Khz
   b. RFID card EM4100
   c. Edimax Switch 8 Port 10/100Mbps
4. Perangkat Keras yang ada di SMKN 1 Katapang
   1. Perangkat Laptop
      a. AMD A8-6410 APU 2.00 Ghz
      b. RAM 4GB
      c. Hardisk 500GB
      d. Monitor LED 16”

Tabel 3.1 Spesifikasi RFID Reader

Weight	70g + USB cable 100g
Size	10.5cm x 7cm x 1.5cm
Interface	USB Full Speed
Reading Distance	5-8cm
Supply Voltage	DC 5V
Supply Current	200mA (Operating); 50mA(standby); 110mA (Normal)

Operating Temperature	-10-70º C
Storage Temperature	-20-80º C
Operating Frequency	125kHz
Smart Card Interface Support	All 125Khz RFID EM4100 EM410X card/tag
Operating System Support	Win 95, Win 98, Win 2000, Win XP, Win Vista, Win 7, Win 8

Tabel 3.2 Spesifikasi RFID tag

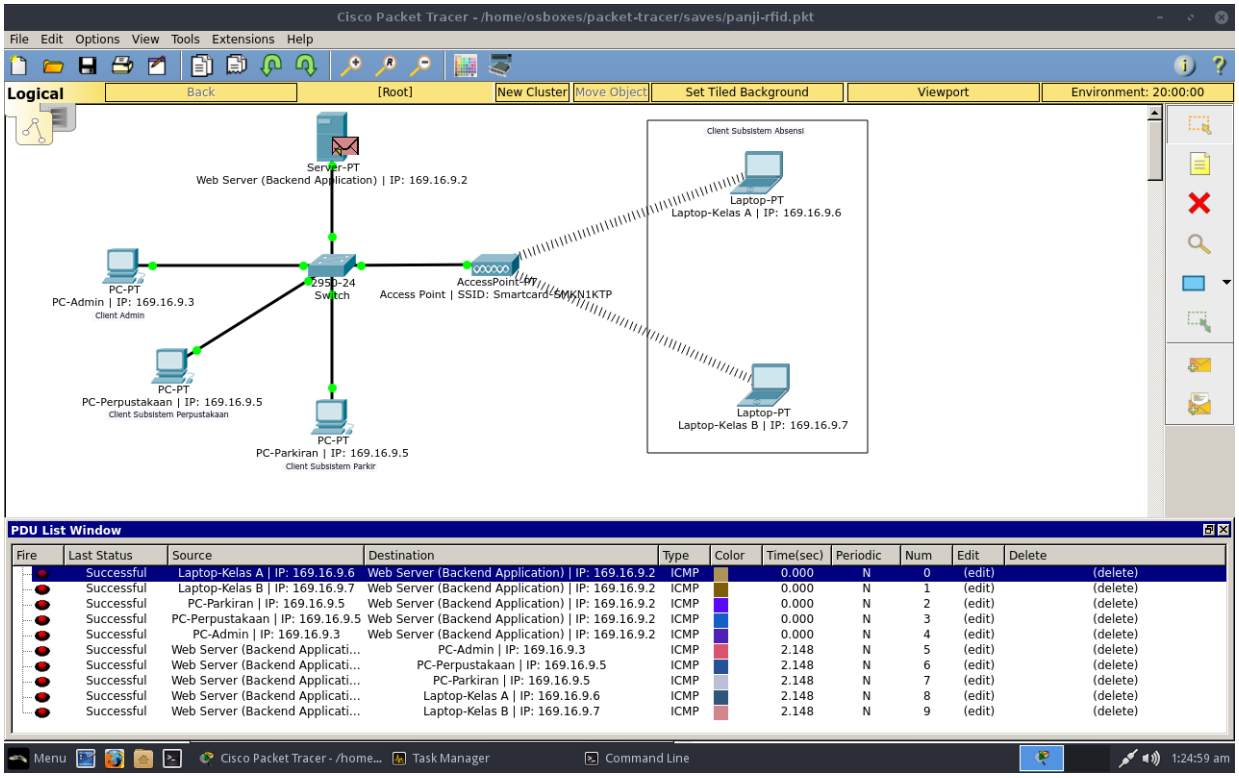
Tag Chip	EM4100
Frekuensi	125Khz
Ukuran ID	64 bit – 224 bit
Temperatur	-20º C -+50º C
Bahan	PVC atau ABS
Dimensions	85.6 × 54 × 0.86 (mm)
Encoding Scheme	Manchester, FSK,PSK,Biphase,NRZ

Dari rincian spesifikasi hardware diatas SMKN 1 Katapang harus menyediakan beberapa perangkat keras untuk mendukung berjalannya sistem yang akan dibangun, antara lain :

1. Perangkat komputer untuk sistem server
2. USB RFID Reader 125Khz
3. Edimax Switch 8 Port 10/100Mbps

3.6.3 Analisis Jaringan *Client-Server*

Pada jaringan client server terdapat sebuah komputer yang berfungsi sebagai server sedangkan komputer-komputer yang lain sebagai client. Server bertugas melayani seluruh komputer yang terdapat dalam jaringan. Sedangkan komputer client berfungsi menerima pelayanan dari komputer server yang setiap komputer client di beri IP address yang sama dengan komputer server.



Gambar 3.13 Client Server

3.6.4 Analisis Pengguna

Analisis kebutuhan perangkat lunak merupakan penguraian kebutuhan-kebutuhan non fungsional yang berhubungan dengan spesifikasi pengguna dan berhubungan dengan perangkat lunak

Tabel 3.3 Pengguna Yang Ada

No	User	Jabatan	Hak Akses	Pengalaman
1.	Guru	Guru/ Wali Kelas	Guru menggunakan aplikasi untuk melengkapi keterangan absen siswa dan melakukan rekapitulasi absensi	Sudah terbiasa menggunakan komputer dan aplikasi seperti MS Office Word
2.	Petugas Parkir	Staf	Petugas parkir menggunakan aplikasi untuk melakukan input data nomor polisi	Sudah terbiasa menggunakan komputer dan aplikasi seperti MS Office Word
3.	Petugas Perpustakaan	Staf	Petugas perpustakaan menggunakan aplikasi untuk melakukan transaksi pinjam dan	Sudah terbiasa menggunakan komputer dan aplikasi

			pengembalian buku	seperti MS Office Word, dan MS Office Excel
4.	Admin	Guru	Admin menggunakan aplikasi untuk mengontrol data master, data pengguna, dan data saldo	Sudah terbiasa menggunakan komputer dan aplikasi seperti MS Office Word, MS Office Excel, dan database
5.	Pelajar	Siswa	Siswa berperan sebagai aktor utama dalam aplikasi ini	Mampu beradaptasi dengan teknologi yang berkembang

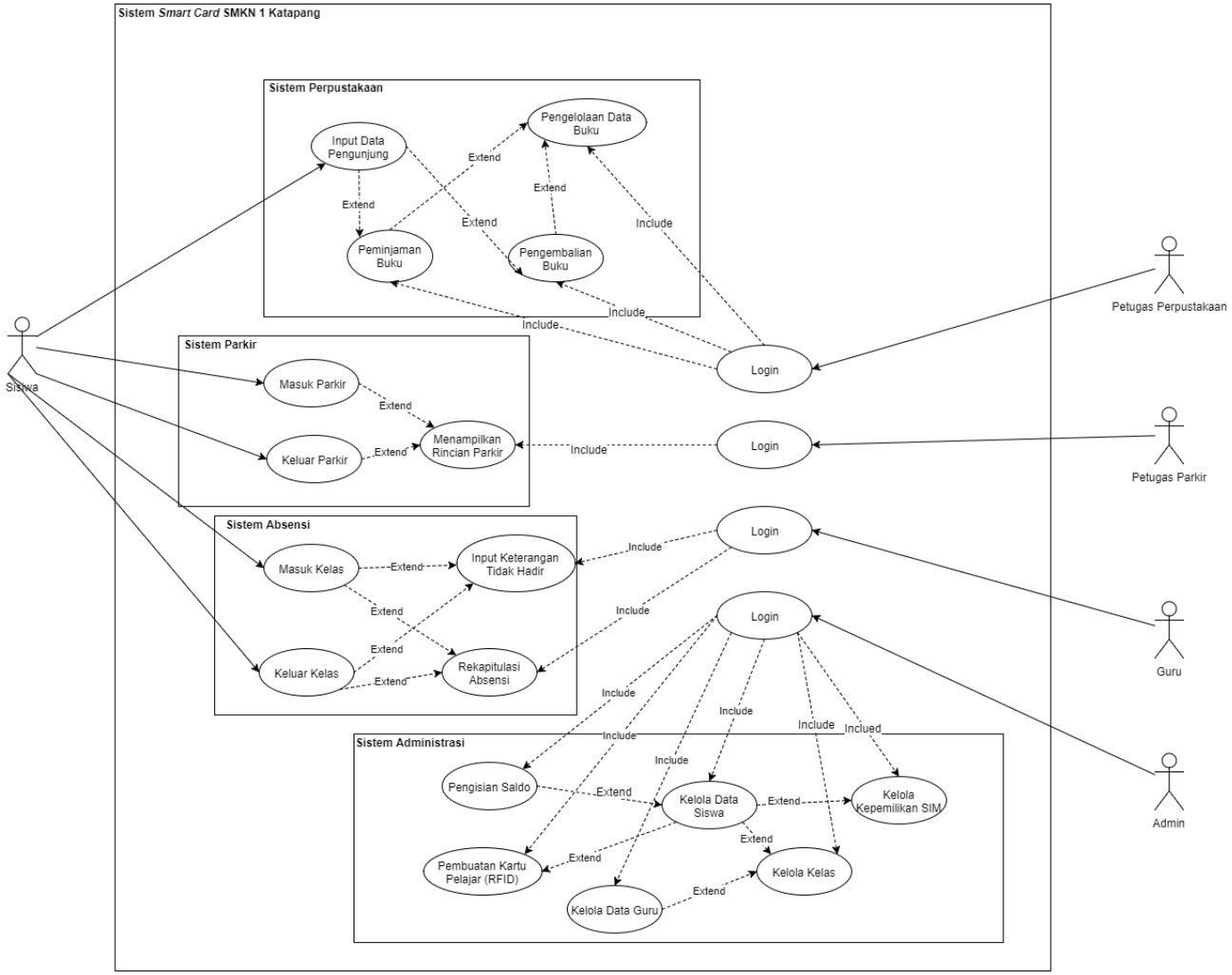
3.7 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional merupakan penguraian dari kebutuhan yang berkaitan dengan fungsi atau proses transformasi yang harus mampu dikerjakan oleh perangkat lunak seperti fitur, karakteristik, dan batasan lainnya yang berhubungan dengan sistem.

3.7.1 Use Case Diagram

Didalam sistem terdapat lima pengguna yaitu Admin, Guru, Petugas Parkir Petugas Perpustakaan dan Siswa. Admin akunting merupakan pengguna yang memiliki hak akses untuk mengelola data master, data siswa, data guru, data kelas dan mengakses data parkir, perpustakaan dan absensi. Petugas parkir merupakan pengguna yang bertugas untuk membaca data RFID ketika siswa parkir dan ketika keluar parkir. Penjaga perpustakaan memiliki tugas memasukan data buku, peminjaman buku, pengembalian buku dan transaksi denda. Sedangkan guru memiliki wewenang untuk menambahkan keterangan terhadap siswa yang tidak hadir dan melakukan rekapitulasi absensi.

Tugas keempat pengguna tersebut sebagai aktor dapat dilihat pada diagram Use Case pada Gambar 3.19.



Gambar 3.14 Use Case Diagram Aplikasi *Smart Card* SMKN 1 Katapang

3.7.1.1 Definisi Aktor

Definisi aktor merupakan penjelasan dari aktor yang ada pada *Use Case Diagram*. Berikut merupakan definisi aktor dari Use Case Diagram pada Gambar 3.4.

Tabel 3.4 Definisi Aktor

No	Aktor	Definisi
1.	Admin	Admin yang memiliki kewenangan untuk mengelola server, data master, dan data pengguna.
2.	Petugas parkir	Petugas yang bertanggung jawab terhadap proses parkir yang ada pada RFID card
3.	Petugas Perpustakaan	Petugas yang mempunyai tanggung jawab terhadap transaksi peminjaman, pengembalian buku dan kelola buku di perpustakaan.
4.	Guru	Petugas yang mempunyai tanggung jawab terhadap absensi kelas.
5.	Siswa	Murid yang bersekolah di sekolah tersebut yang memiliki kartu pelajar pintar untuk identitas diri sebagai siswa SMKN 1 Katapang, keamanan parkir kendaraan, meminjam buku di perpustakaan, dan untuk absensi kehadiran.

3.7.1.2 Definisi Use Case Diagram

Definisi *Use Case* merupakan penjelasan dari fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. Berikut merupakan definisi *Use Case* dari gambar 3.5.

Tabel 3.5 Definisi Use Case Diagram

No	Use Case	Definisi
1.	Login	Proses login sebagai autentifikasi kewenangan hak akses pengguna.
2.	Kelola Data Siswa	Memiliki fungsional menampilkan data dan mengelola data siswa
3.	Kelola Data Guru	Memiliki fungsional menampilkan dan mengelola data guru
4.	Kelola Data Kelas	Memiliki fungsional menampilkan dan mengelola data Kelas
5.	Top Up Saldo	Memiliki fungsional untuk menambahkan saldo.
6.	Registasi Kartu	Memiliki fungsional untuk registrasi kartu pelajar menggunakan RFID .
7.	Registrasi SIM	Memiliki fungsional untuk memasukan data kepemilikan SIM.

8.	Input Keterangan Tidak Hadir	Memiliki fungsional untuk melengkapi absensi siswa dan memberikan keterangan mengenai kehadiran siswa.
9.	Rekapitulasi Absensi	Memiliki fungsional untuk melakukan rekapitulasi absensi.
10.	Menampilkan Rincian Parkir	Memiliki fungsional untuk menampilkan rincian parkir.
11.	Pengelolaan Data Buku	Memiliki fungsional untuk memasukan data buku untuk koleksi perpustakaan.
12.	Peminjaman Buku	Memiliki fungsional untuk memasukan data buku yang dipinjam oleh siswa.
13.	Pengembalian Buku	Memiliki fungsional untuk memeriksa data buku yang telah dipinjam oleh siswa.
14.	Isi data pengunjung	Memiliki fungsional untuk mencatat data pengunjung perpustakaan
15.	Masuk Parkir	Memiliki fungsional untuk mencatat data parkir saat datang
16.	Keluar Parkir	Memiliki fungsional untuk mencatat data parkir saat keluar
17.	Masuk Kelas	Memiliki fungsional untuk mencatat data absen saat datang
18.	Keluar Kelas	Memiliki fungsional untuk mencatat data absen saat pulang

3.7.1.3 Skenario Use Case Diagram

Skenario *Use Case* mendeskripsikan langkah-langkah dalam proses bisnis, baik yang dilakukan aktor terhadap sistem maupun yang dilakukan oleh sistem terhadap aktor.

a. Skenario Use Case Login

Skenario use case login menampilkan menu login untuk menampilkan menu utama data masing masing hak akses.

skenario use case menampilkan materi seperti pada tabel 3.6.

Tabel 3.6 Skenario Use Case Login

Use case Name	***Login***
Related Requirement	Sistem menyediakan form login
Goal In Context	Administrator, Guru, Petugas Pakir atau Petugas Perpustakaan berhasil login kedalam sistem
Preconditions	Sistem menampilkan halaman login
Succesfull End Condition	Login berhasil

Failed End Condition	Login gagal	
Primary Actors	Administrator, Guru, Petugas Pakir dan Petugas Perpustakaan	
Secondary Actors	None	
Trigger	Sistem mengakses data pengguna pada database	
Main Flow	***Step***	***Action***
	1	Administrator menginputkan username
	2	Administrator menginputkan password
	3	Admin menekan tombol login
	4	Sistem menampilkan pesan login berhasil
	5	Sistem menampilkan halaman administrator
Extensions	***Step***	***Action***
	4.1	Sistem menampilkan pesan login gagal
	5.1	Sistem tidak menampilkan halaman administrator

b. Skenario *Use Case* Kelola Data Siswa

Tabel 3.7 Use Case Kelola Data Siswa

Use case Name	**Kelola Data Siswa**	
Related Requirement	Sistem menyediakan form kelola data siswa	
Goal In Context	Administrator dapat mengelola data siswa	
Preconditions	Sistem menampilkan halaman kelola data siswa	
Succesfull End Condition	Kelola data berhasil	
Failed End Condition	Kelola data gagal	
Primary Actors	Administrator	
Secondary Actors	None	
Trigger	Sistem mengakses data siswa pada database	
Main Flow	***Step***	***Action***
	1	Sistem menampilkan data siswa
	2	Administrator menambah data siswa baru
	3	Administrator menginputkan data berkaitan dengan data siswa

	4	Administrator menekan tombol tambah
	5	Sistem menampilkan pesan berhasil
Extensions	***Step***	***Action***
	2.1	Administrator mengedit data siswa
	2.2	Administrator menghapus data siswa
	5.1	Sistem menampilkan pesan gagal

c. Skenario *Use Case* Kelola Data Guru

Tabel 3.8 Use Case Kelola Data Guru

Use case Name	**Kelola Data Guru**	
Related Requirement	Sistem menyediakan form kelola data guru	
Goal In Context	Administrator dapat mengelola data guru	
Preconditions	Sistem menampilkan halaman kelola data guru	
Succesfull End Condition	Kelola data berhasil	
Failed End Condition	Kelola data gagal	
Primary Actors	Administrator	
Secondary Actors	None	
Trigger	Sistem mengakses data guru pada database	
Main Flow	***Step***	***Action***
	1	Sistem menampilkan data guru
	2	Administrator menambah data guru baru
	3	Administrator menginputkan data berkaitan dengan data guru
	4	Administrator menekan tombol tambah
	5	Sistem menampilkan pesan berhasil
Extensions	***Step***	***Action***
	2.1	Administrator mengedit data guru
	2.2	Administrator menghapus data guru
	5.1	Sistem menampilkan pesan gagal

d. Skenario *Use Case* Kelola Data Kelas

Tabel 3.9 Use Case Kelola Data Kelas

Use case Name	**Kelola Data Kelas**	
Related Requirement	Sistem menyediakan form kelola data kelas	
Goal In Context	Administrator dapat mengelola data kelas	
Preconditions	Sistem menampilkan halaman kelola data kelas	
Succesfull End Condition	Kelola data berhasil	
Failed End Condition	Kelola data gagal	
Primary Actors	Administrator	
Secondary Actors	None	
Trigger	Sistem mengakses data kelaspada database	
Main Flow	***Step***	***Action***
	1	Sistem menampilkan data kelas
	2	Administrator menambah data kelasbaru
	3	Administrator menginputkan data berkaitan dengan data kelas
	4	Administrator menekan tombol tambah
	5	Sistem menampilkan pesan berhasil
Extensions	***Step***	***Action***
	2.1	Administrator mengedit data kelas
	2.2	Administrator menghapus data kelas
	5.1	Sistem menampilkan pesan gagal

e. Skenario *Use Case* Top Up Saldo

Tabel 3.10 Use Case Input Top Up Saldo

Use case Name	**Top Up Saldo**
Related Requirement	Sistem menyediakan form top up saldo
Goal In Context	Saldo akhir bertambah
Preconditions	Sistem menampilkan halaman top up saldo
Succesfull End Condition	Jumlah saldo akhir bertambah dari sebelumnya
Failed End Condition	Jumlah saldo akhir tidak bertambah

Primary Actors	Administrator	
Secondary Actors	None	
Trigger	Sistem mengakses data saldo pada database	
Main Flow	***Step***	***Action***
	1	Sistem memindai *Smartcard*
	2	Sistem mencari data saldo sesuai dengan smart card
	3	Sistem menampilkan saldo akhir dari siswa yang *Smartcard* telah dipindai
	4	Admin menginput jumlah saldo yang akan di top up
	5	Admin menekan tombol top up
	6	Sistem menampilkan pesan top up berhasil
Extensions	***Step***	***Action***
	3.1	Sistem menampilkan pesan data tidak ditemukan
	6.1	Sistem menampilkan pesan top up gagal

f. Seknario Use Case Membuat Kartu

Tabel 3.11 Use Case Buat Kartu

Use case Name	**Buat Kartu**	
Related Requirement	Sistem menyediakan form pembuatan kartu	
Goal In Context	Kartu baru terintegrasi dengan siswa yang bersangkutan	
Preconditions	Sistem menampilkan halaman pembuatan kartu	
Succesfull End Condition	Data siswa terintegrasi dengan *Smartcard*	
Failed End Condition	Data siswa tidak terintegrasi dengan *Smartcard*	
Primary Actors	Administrator	
Secondary Actors	None	
Trigger	Sistem mengupdate data siswa dan data *Smartcard* pada database	
Main Flow	***Step***	***Action***
	1	Administrator memindai *smartcard*
	2	Sistem mencari data siswa sesuai dengan inputan
	3	Sistem menampilkan data siswa

	4	Admin menekan tombol Register Kartu
	5	menginputkan nomor induk siswa atau nama siswa
	6	Sistem membaca *Smartcard*
	7	Sistem menampilkan pesan *Smartcard* valid
	8	Admin menekan tombol Simpan
	9	Sistem menambah data smart card
	10	Sistem mengupdate kolom rfid pada tabel siswa
	11	Sistem menampilkan pesan pembuatan kartu baru berhasil
Extensions	***Step***	***Action***
	3.1	Sistem menampilkan pesan siswa tidak ditemukan
	7.1	Sistem menampilkan pesan SmartdCard tidak valid
	11.1	Sistem menampilkan pesan pembuatan kartu baru gagal

g. Skenario Use Case Kepemilikan SIM

Tabel 3.12 Use Case Kepemilikan SIM

Use case Name	**Kepemilikan SIM**	
Related Requirement	Sistem menyediakan form pendaftaran kepemilikan SIM	
Goal In Context	SIM yang dimiliki siswa terdaftar pada sistem	
Preconditions	Sistem menampilkan halaman pendaftaran kepemilikan SIM	
Succesfull End Condition	Pendaftaran SIM berhasil	
Failed End Condition	Pendaftaran SIM gagal	
Primary Actors	Administrator	
Secondary Actors	None	
Trigger	Sistem mengupdate data siswa pada database	
Main Flow	***Step***	***Action***
	1	Administrator memindai *Smartcard*
	2	Sistem mencari data saldo sesuai dengan *Smartcard*
	3	Sistem menampilkan data status kepemilikan SIM untuk siswa tersebut
	4	Admin menekan tombol update
	5	Admin mengubah status kepemilikan SIM menjadi YA

	6	Admin menekan tombol simpan
	7	Sistem mengupdate kolom punya_sim pada tabel siswa
	8	Sistem menampilkan pesan pendaftaran SIM berhasil
Extensions	***Step***	***Action***
	3.1	Sistem menampilkan pesan data tidak ditemukan
	8.1	Sistem menampilkan pesan pendaftaran SIM gagal

h. Skenario Use Case Input Keterangan Tidak Hadir

Tabel 3.13 Use Case Input Keterangan Tidak Hadir

Use case Name	**Input Keterangan Tidak Hadir**	
Related Requirement	Sistem menyediakan form keterangan tidak hadir	
Goal In Context	Guru berhasil mengelola data keterangan tidak hadir	
Preconditions	Sistem menampilkan halaman keterangan tidak hadir	
Succesfull End Condition	Input keterangan tidak hadir berhasil	
Failed End Condition	Input keterangan tidak hadir gagal	
Primary Actors	Guru	
Secondary Actors	None	
Trigger	Sistem mengupdate data sakit, izin, tanpa_keterangan pada database	
Main Flow	***Step***	***Action***
	1	Sistem menampilkan data absensi
	2	Guru memilih update absensi
	3	Sistem menampilkan pilihan tidak hadir dan kolom keterangan
	4	Guru memilih pilihan tidak hadir dan mengisi form keterangan
	5	Guru menekan tombol update
	6	Sistem menampilkan pesan update absensi berhasil
Extensions	***Step***	***Action***
	3.1	Guru memilih sakit dan mengisi keterangan
	3.2	Guru memilih izin dan mengisi keterangan
	3.3	Guru memilih alfa dan mengisi keterangan
	6.1	Sistem menampilkan pesan update absensi gagal

i. Skenario *Use Case* Rekapitulasi Absensi

Tabel 3.14 Use Case Rekapitulasi Absensi

Use case Name	**Rekapitulasi Absensi**	
Related Requirement	Sistem menyediakan form rekapitulasi absensi	
Goal In Context	Guru berhasil merekapitulasi absensi	
Preconditions	Sistem menampilkan halaman rekapitulasi absensi	
Succesfull End Condition	Data rekapitulasi absensi berhasil	
Failed End Condition	Data rekapitulasi absensi gagal	
Primary Actors	Guru	
Secondary Actors	None	
Trigger	Sistem mengupdate rekap_absensi pada database	
Main Flow	***Step***	***Action***
	1	Sistem menampilkan data absensi
	2	Guru memilih rekapitulasi absensi
	3	Sistem menampilkan data absensi dan pilihan untuk rekapilasi
	4	Guru memilih pilihan rekapitulasi absensi
	5	Guru menekan rekapitulasi absensi
	6	Sistem menampilkan pesan rekapitulasi absensi berhasil
Extensions	***Step***	***Action***
	3.1	Guru memilih kelas yang akan direkapitulasi
	6.1	Sistem menampilkan pesan update absensi gagal

j. Skenario *Use Case* Input Nomor Polisi

Tabel 3.15 Use Case Input Nomor Polisi

Use case Name	**Input Nomor Polisi**
Related Requirement	Sistem menyediakan form input nomor polisi
Goal In Context	Petugas parkir berhasil menginput nomor polisi
Preconditions	Sistem menampilkan halaman input nomor polisi
Succesfull End Condition	Input nomor polisi berhasil
Failed End Condition	Input nomor polisi gagal

Primary Actors	Petugas parkir	
Secondary Actors	None	
Trigger	Sistem mengupdate data parkir pada database	
Main Flow	***Step***	***Action***
	1	Sistem menampilkan data parkir
	2	Petugas parkir memilih input nomor polisi
	3	Sistem menampilkan form input nomor polisi
	4	Petugas parkir menginput nomor polisi
	5	Petugas parkir menekan selesai
	6	Sistem menampilkan input nomor polisi berhasil
Extensions	***Step***	***Action***
	6.1	Sistem menampilkan pesan input nomor polisi gagal

k. Skenario *Use Case* Menampilkan Rincian Parkir

Tabel 3.16 Use Case Menampilkan Rincian Parkir

Use case Name	**Menampilkan Rincian Parkir**	
Related Requirement	Sistem menyediakan form rincian parkir	
Goal In Context	Petugas parkir berhasil melihat rincian parkir	
Preconditions	Sistem menampilkan halaman rincian parkir	
Succesfull End Condition	Rincian parkir berhasil dilihat	
Failed End Condition	Rincian parkir tidak ada	
Primary Actors	Petugas parkir	
Secondary Actors	None	
Trigger	Sistem mengupdate data parkir pada database	
Main Flow	***Step***	***Action***
	1	Sistem menampilkan data parkir
	2	Petugas parkir rincian parkir
	3	Sistem menampilkan rincian parkir

l. Skenario *Use Case* Pengelolaan Data Buku

Tabel 3.17 Use Case Pengelolaan Data Buku

Use case Name	**Pengelolaan Data Buku**	
Related Requirement	Sistem menyediakan form kelola data buku	
Goal In Context	Petugas perpustakaan berhasil mengelola data buku	
Preconditions	Sistem menampilkan halaman kelola data buku	
Succesfull End Condition	Input data buku berhasil	
Failed End Condition	Input data buku gagal	
Primary Actors	Petugas perpustakaan	
Secondary Actors	None	
Trigger	Sistem mengupdate buku_perpustakaan pada database	
Main Flow	***Step***	***Action***
	1	Sistem menampilkan data buku
	2	Petugas perpustakaan menambah data buku
	3	Sistem menampilkan form kelola data buku
	4	Petugas perpustakaan menginput data buku
	5	Petugas perpustakaan menekan selesai
	6	Sistem menampilkan input data buku berhasil
Extensions	***Step***	***Action***
	3.1	Petugas perpustakaan mengedit data buku
	3.2	Petugas perpustakaan menghapus data buku
	6.1	Sistem menampilkan pesan input data buku gagal

m. Skenario *Use Case* Peminjaman Buku

Tabel 3.18 Use Case Peminjaman Buku

Use case Name	**Peminjaman Buku**
Related Requirement	Sistem menyediakan form peminjaman buku
Goal In Context	Petugas parkir berhasil menginput peminjaman buku
Preconditions	Sistem menampilkan halaman input peminjaman buku
Succesfull End Condition	Input peminjaman buku berhasil
Failed End Condition	Input peminjaman buku gagal

Primary Actors	Petugas perpustakaan	
Secondary Actors	None	
Trigger	Sistem mengupdate data detail_peminjaman pada database	
Main Flow	***Step***	***Action***
	1	Sistem menampilkan data peminjaman buku
	2	Petugas perpustakaan memilih peminjaman buku
	3	Sistem menampilkan form input peminjaman buku
	4	Petugas perpustakaan memindai *Smartcard*
	5	Petugas perpustakaan menginput peminjaman buku
	6	Petugas perpustakaan menekan selesai
	7	Sistem menampilkan input peminjaman buku berhasil
Extensions	***Step***	***Action***
	7.1	Sistem menampilkan pesan input peminjaman buku gagal

n. Skenario *Use Case* Pengembalian Buku

Tabel 3.19 Use Case Pengembalian Buku

Use case Name	**Pengembalian Buku**	
Related Requirement	Sistem menyediakan form pengembalian buku	
Goal In Context	Petugas parkir berhasil menginput pengembalian buku	
Preconditions	Sistem menampilkan halaman input pengembalian buku	
Succesfull End Condition	Input pengembalian buku berhasil	
Failed End Condition	Input pengembalian buku gagal	
Primary Actors	Petugas perpustakaan	
Secondary Actors	None	
Trigger	Sistem mengupdate data detail_peminjaman pada database	
Main Flow	***Step***	***Action***
	1	Sistem menampilkan data pengembalian buku
	2	Petugas perpustakaan memilih pengembalian buku
	3	Sistem menampilkan form input pengembalian buku
	4	Petugas perpustakaan memindai *Smartcard*

	5	Petugas perpustakaan mengecek pengembalian buku
	6	Petugas perpustakaan menekan selesai
	7	Sistem menampilkan input pengembalian buku berhasil
Extensions	***Step***	***Action***
	7.1	Sistem menampilkan pesan input nomor polisi gagal

o. Skenario *Use Case* Masuk Parkir

Tabel 3.20 Use Case Masuk Parkir

Use case Name	**Masuk Parkir**	
Related Requirement	Sistem menyediakan form Masuk Parkir	
Goal In Context	Waktu siswa masuk berhasil diinput	
Preconditions	Sistem menampilkan halaman parkir	
Succesfull End Condition	Waktu siswa masuk berhasil diupdate	
Failed End Condition	Waktu siswa masuk gagal diupdate	
Primary Actors	Siswa	
Secondary Actors	None	
Trigger	Sistem mengupdate data parkir pada database	
Main Flow	***Step***	***Action***
	1	Sistem menampilkan data parkir
	2	Siswa memindai *Smartcard*
	3	Sistem menginput waktu siswa masuk
	4	Sistem menampilkan input waktu siswa masuk berhasil
Extensions	***Step***	***Action***
	4.1	Sistem menampilkan pesan waktu siswa masuk gagal

p. Skenario *Use Case* Keluar Parkir

Tabel 3.21 Use Case Keluar Parkir

Use case Name	**Keluar Parkir**
Related Requirement	Sistem menyediakan form keluar Parkir

Goal In Context	Waktu siswa keluar berhasil diinput	
Preconditions	Sistem menampilkan halaman parkir	
Succesfull End Condition	Waktu siswa keluar berhasil diupdate	
Failed End Condition	Waktu siswa keluar gagal diupdate	
Primary Actors	Siswa	
Secondary Actors	None	
Trigger	Sistem mengupdate data parkir pada database	
Main Flow	***Step***	***Action***
	1	Sistem menampilkan data parkir
	2	Siswa memindai *Smartcard*
	3	Sistem menginput waktu siswa keluar
	4	Sistem menampilkan input waktu siswa keluar berhasil
Extensions	***Step***	***Action***
	4.1	Sistem menampilkan pesan waktu siswa keluar gagal

q. Skenario *Use Case* Masuk Kelas

Tabel 3.22 Use Case Masuk Kelas

Use case Name	**Masuk Kelas**	
Related Requirement	Sistem menyediakan form masuk kelas	
Goal In Context	Waktu siswa kelas berhasil diinput	
Preconditions	Sistem menampilkan halaman absensi	
Succesfull End Condition	Waktu siswa masuk berhasil diupdate	
Failed End Condition	Waktu siswa masuk gagal diupdate	
Primary Actors	Siswa	
Secondary Actors	None	
Trigger	Sistem mengupdate data absensi pada database	
Main Flow	***Step***	***Action***
	1	Sistem menampilkan data absensi
	2	Siswa memindai *Smartcard*
	3	Sistem menginput waktu siswa masuk

	4	Sistem menampilkan input waktu siswa masuk berhasil
Extensions	***Step***	***Action***
	4.1	Sistem menampilkan pesan waktu siswa masuk gagal

r. Skenario *Use Case* Keluar Kelas

Tabel 3.23 Use Case Keluar Kelas

Use case Name	**Keluar Kelas**	
Related Requirement	Sistem menyediakan form keluar kelas	
Goal In Context	Waktu siswa keluar berhasil diinput	
Preconditions	Sistem menampilkan halaman absensi	
Succesfull End Condition	Waktu siswa keluar berhasil diupdate	
Failed End Condition	Waktu siswa keluar gagal diupdate	
Primary Actors	Siswa	
Secondary Actors	None	
Trigger	Sistem mengupdate data absensi pada database	
Main Flow	***Step***	***Action***
	1	Sistem menampilkan data absensi
	2	Siswa memindai *Smartcard*
	3	Sistem menginput waktu siswa keluar
	4	Sistem menampilkan input waktu siswa keluar berhasil
Extensions	***Step***	***Action***
	4.1	Sistem menampilkan pesan waktu siswa keluar gagal

s. Skenario *Use Case* Isi Data Pengunjung

Tabel 3.24 Use Case Isi Data Pengunjung

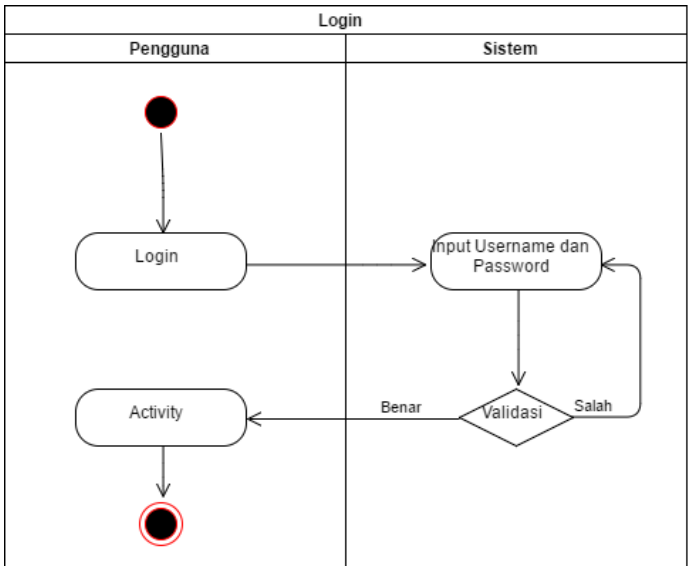
Use case Name	**Isi Data Pengunjung**
Related Requirement	Sistem menyediakan data pengunjung
Goal In Context	Waktu siswa berkunjung ke perpustakaan tercatat
Preconditions	Sistem menampilkan isi pengunjung

Succesfull End Condition	Waktu siswa berkunjung berhasil diupdate	
Failed End Condition	Waktu siswa berkunjung gagal diupdate	
Primary Actors	Siswa	
Secondary Actors	None	
Trigger	Sistem mengupdate data pengunjung pada database	
Main Flow	***Step***	***Action***
	1	Sistem menampilkan data pengunjung
	2	Siswa memindai *Smartcard*
	3	Sistem menginput waktu siswa berkunjung
	4	Sistem menampilkan input waktu siswa berkunjung berhasil
Extensions	***Step***	***Action***
	4.1	Sistem menampilkan pesan waktu siswa berkunjung gagal

3.7.2 Activity Diagram

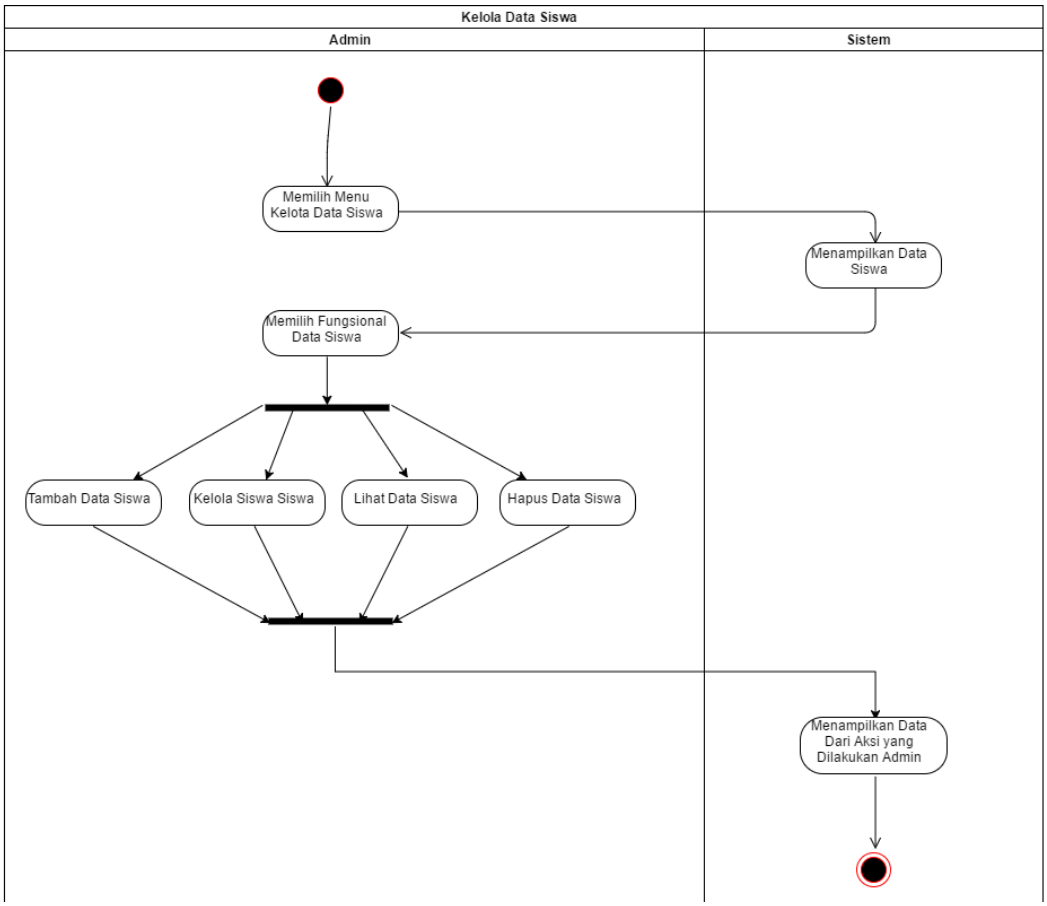
Activity Diagram mendeskripsikan aliran kerja (*workflow*) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. *Activity* diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

a. *Activity* Diagram Login



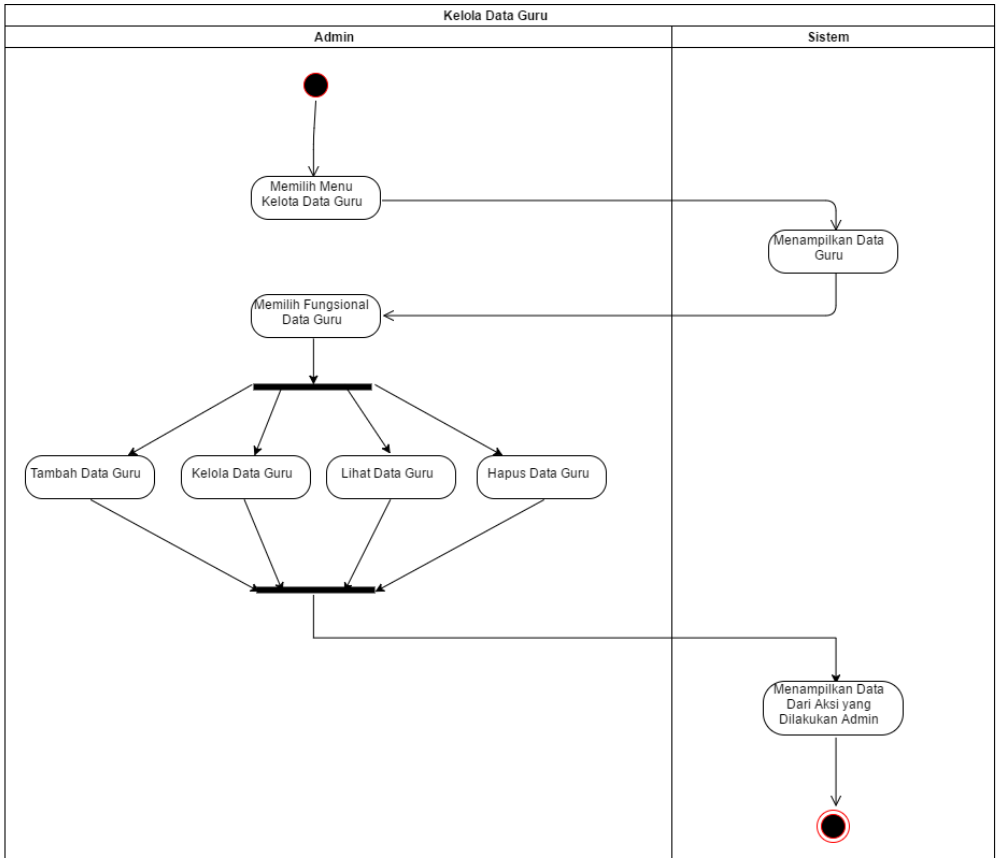
Gambar 3.15 Activity Diagram Login

b. *Activity* Diagram Kelola Data Siswa



Gambar 3.16 Activity Diagram Kelola Data Siswa

c. *Activity* Diagram Kelola Data Guru



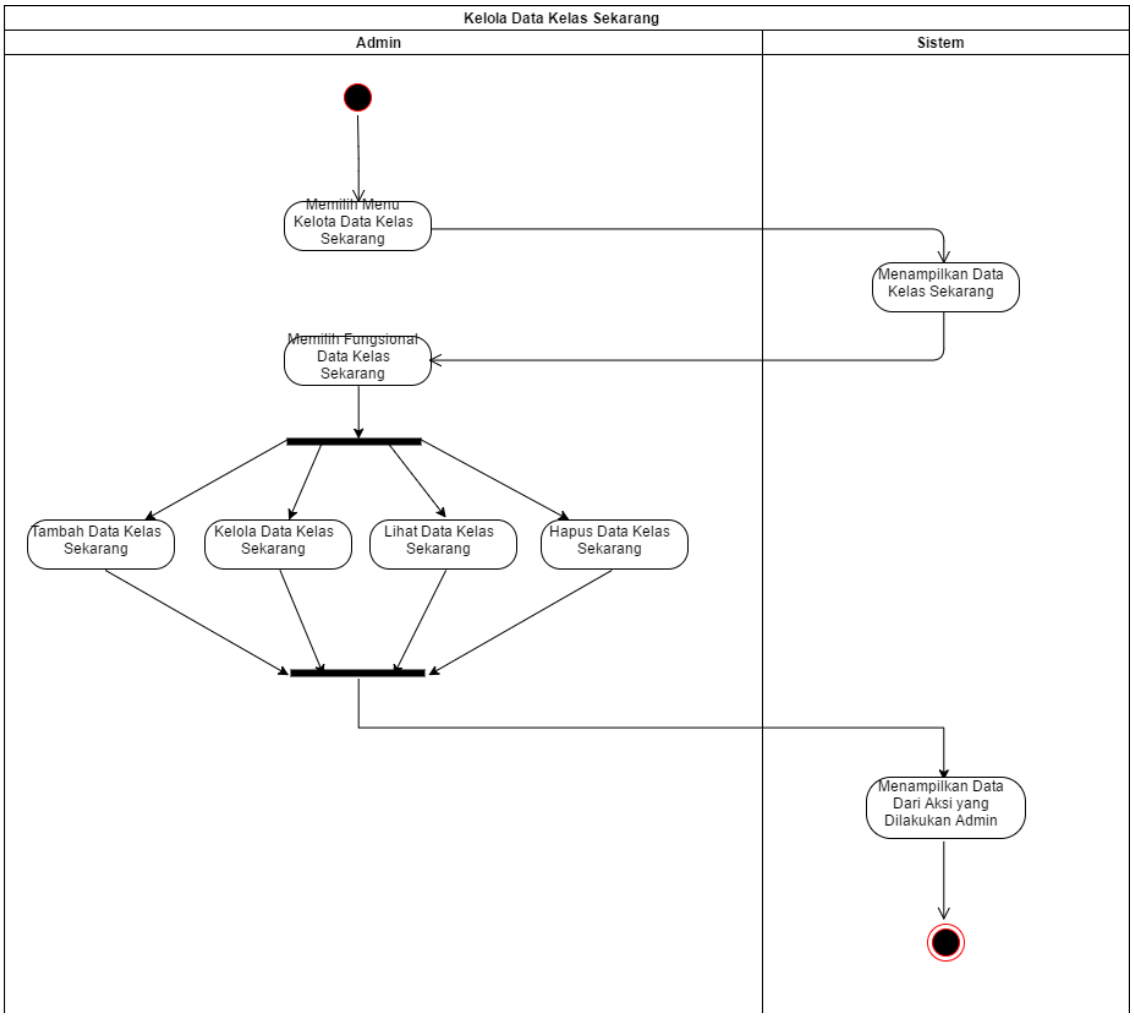
Gambar 3.17 Activity Diagram Kelola Data Guru

d. *Activity* Diagram Data Kelas



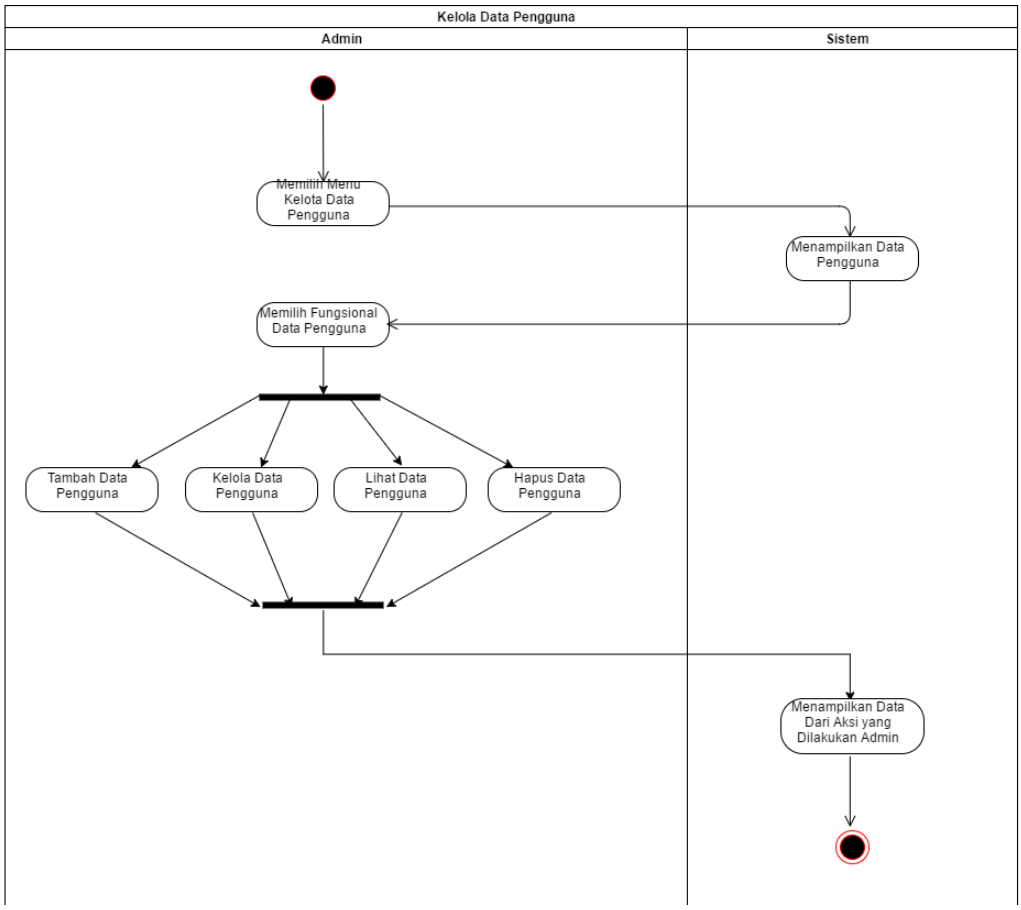
Gambar 3.18 Activity Diagram Kelola Data Kelas

e. *Activity* Diagram Kelola Data Kelas Sekarang



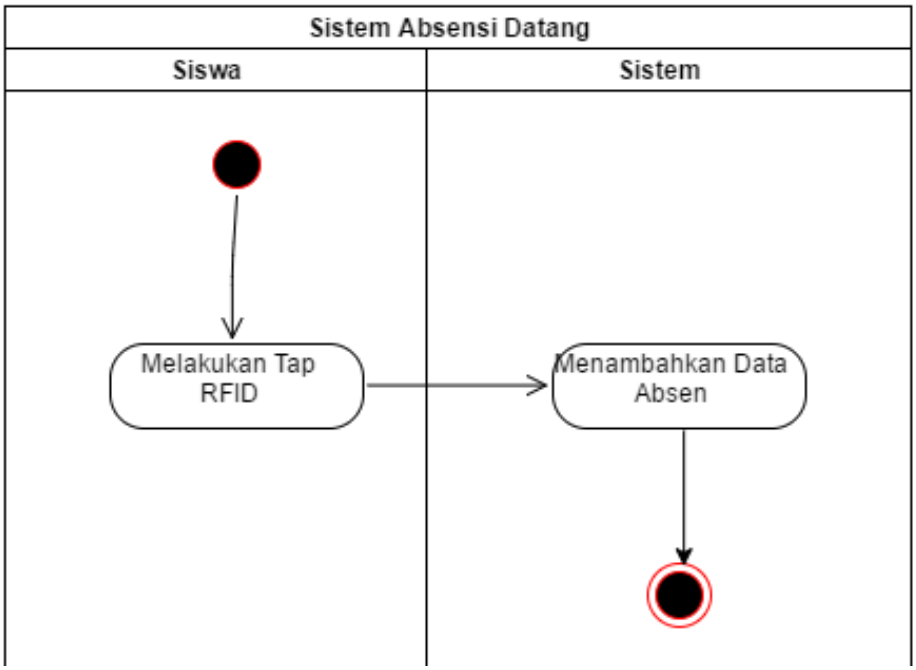
Gambar 3.19 Activity Diagram Kelola Data Kelas Sekarang

f. *Activity* Diagram Kelola Data Pengguna



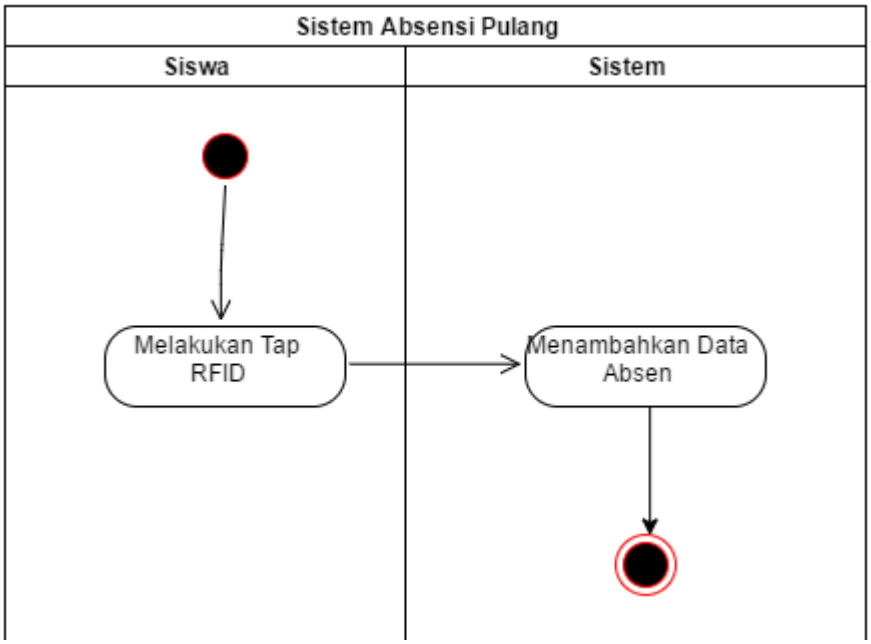
Gambar 3.20 Activity Diagram Kelola Data Pengguna

g. *Activity* Diagram Absensi Masuk



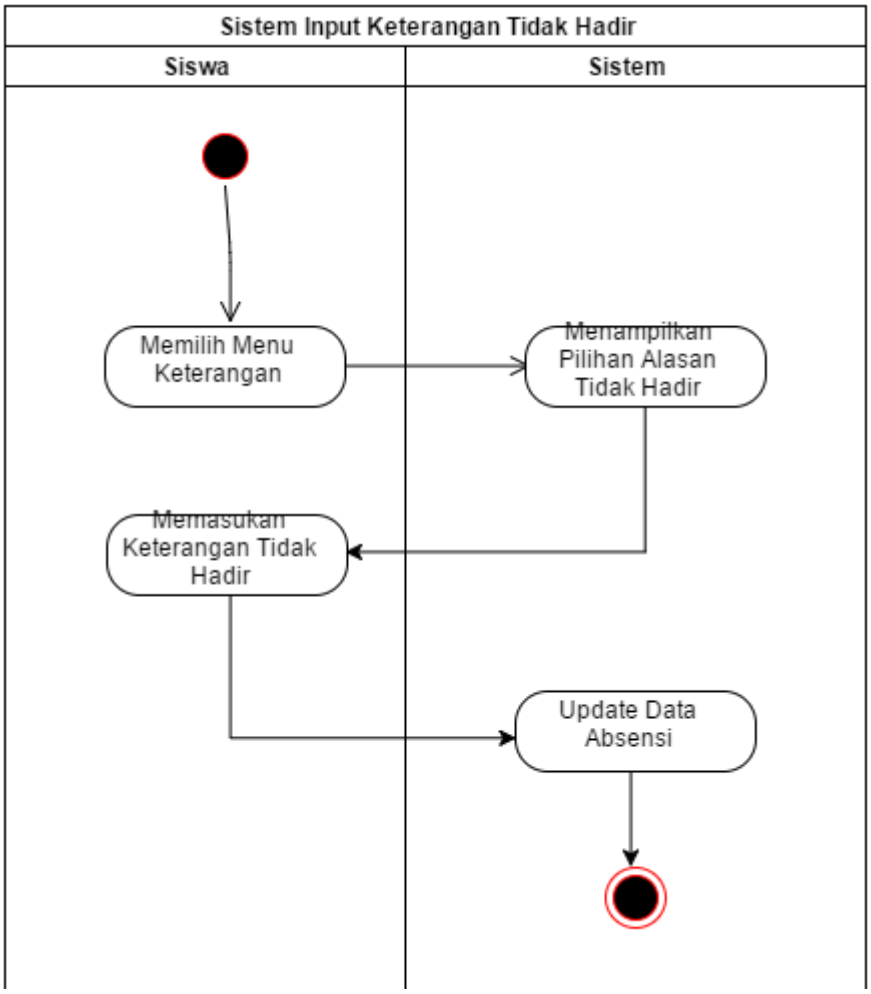
Gambar 3.21 Activity Diagram Absensi Datang

h. *Activity* Diagram Absensi Pulang



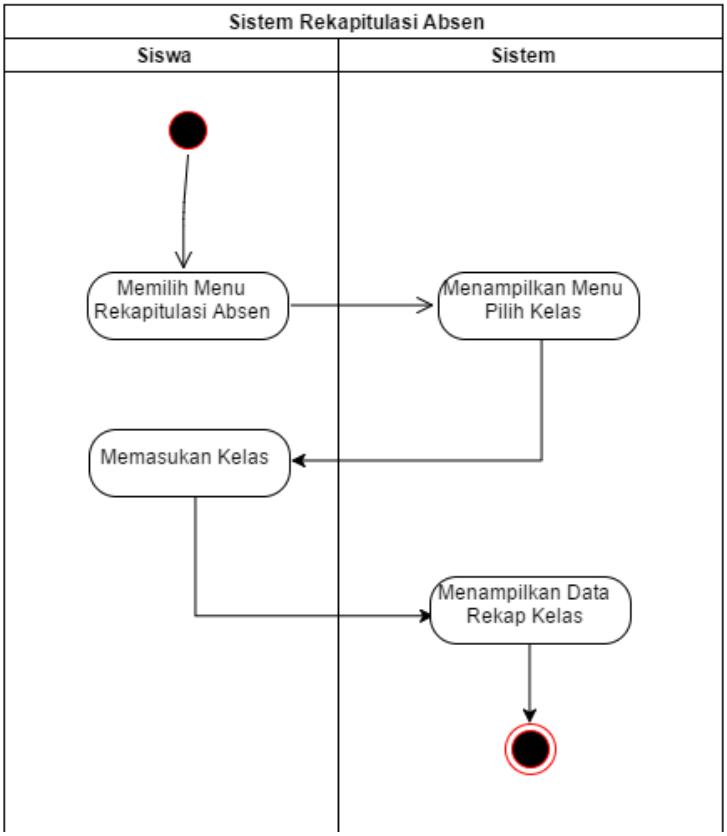
Gambar 3.22 Activity Diagram Absensi Pulang

i. *Activity* Diagram Mengisi Keterangan



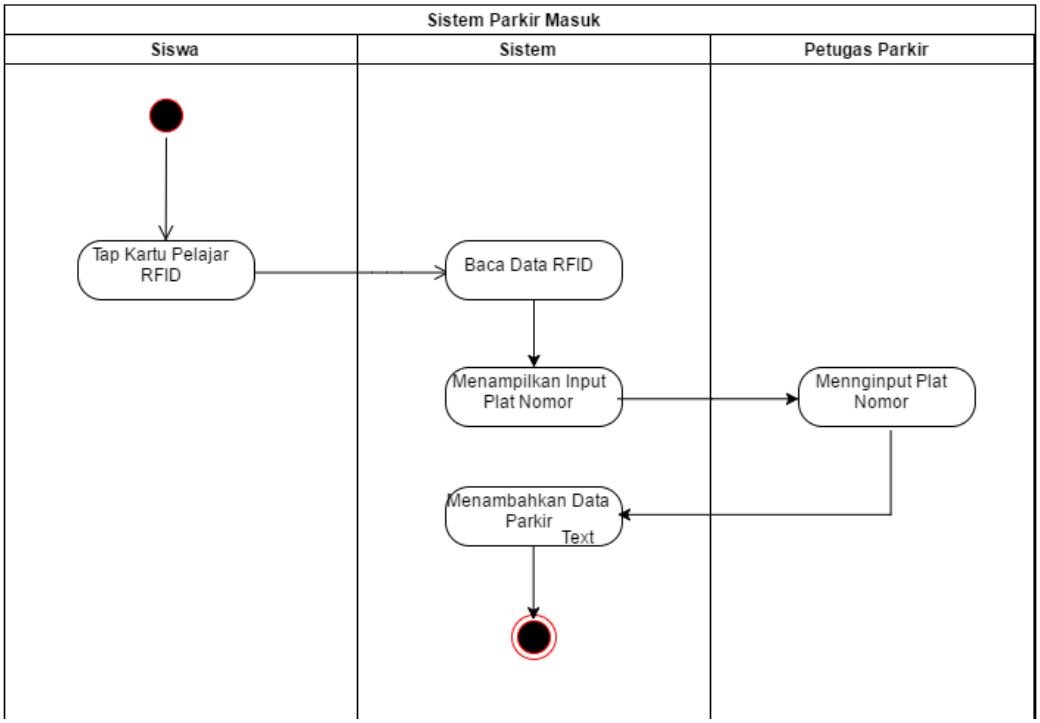
Gambar 3.23 Activity Diagram Keterangan Tidak Hadir

j. A*ctivity* Diagram Rekapitulasi Absensi



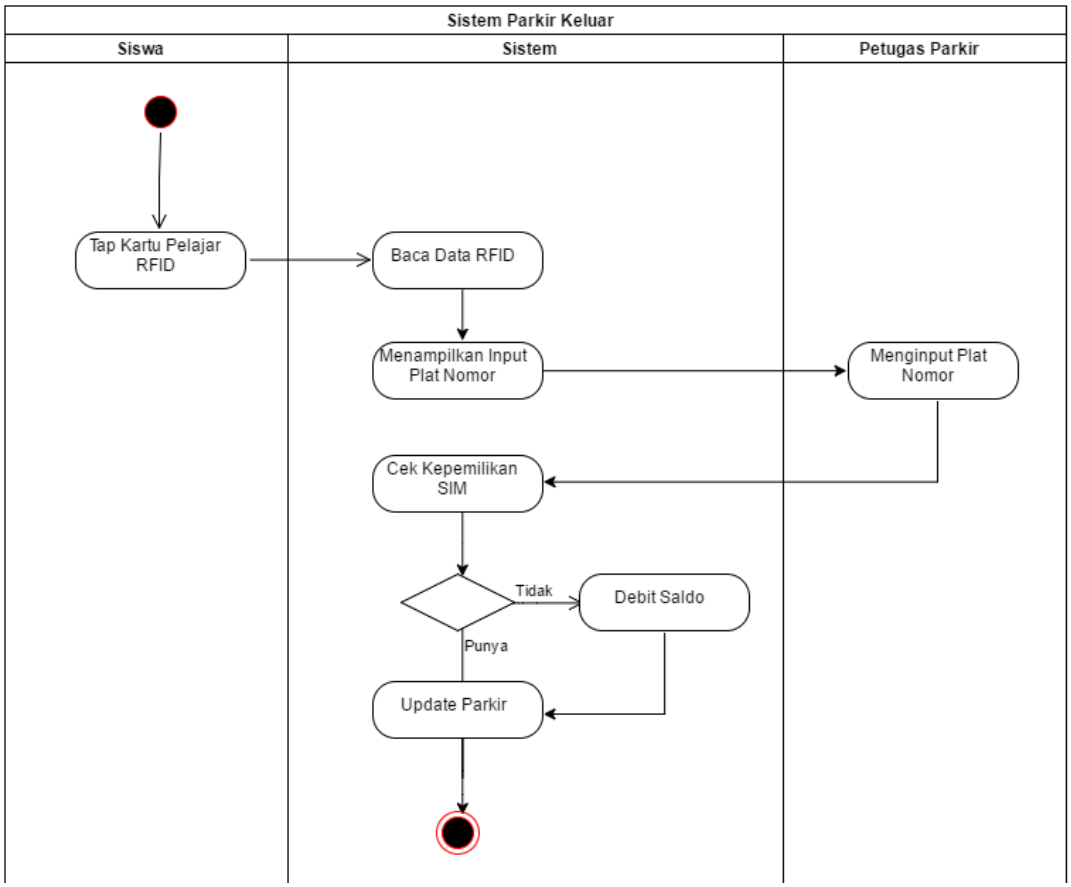
Gambar 3.24 Activity Diagram Rekapitulasi Absensi

k. *Activity* Diagram Masuk Parkir



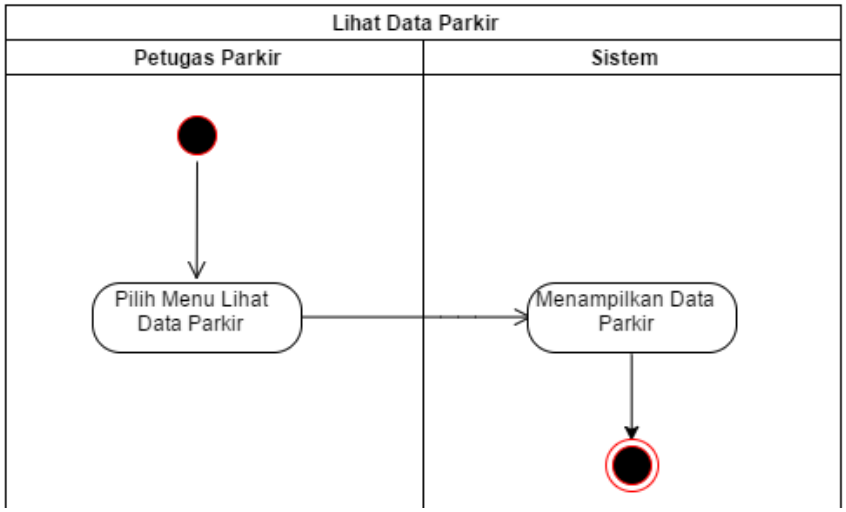
Gambar 3.25 Activity Diagram Masuk Parkir

l. *Activity* Diagram Keluar Parkir



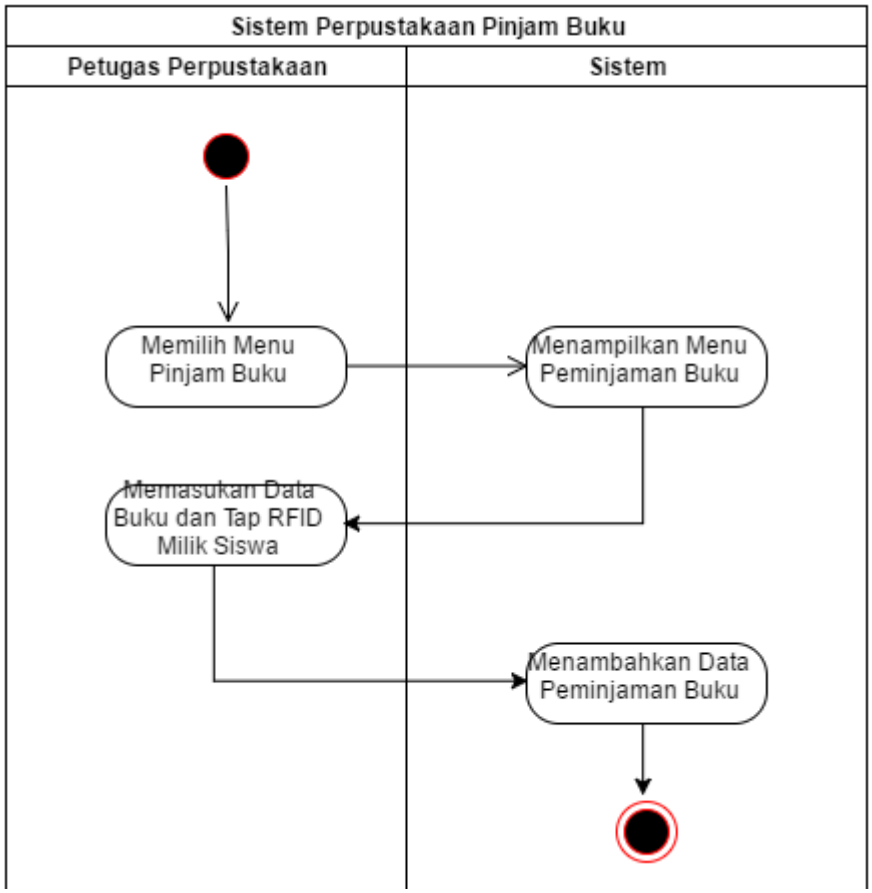
Gambar 3.26 Activity Diagram Keluar Parkir

m. *Activity* Diagram Rincian Parkir



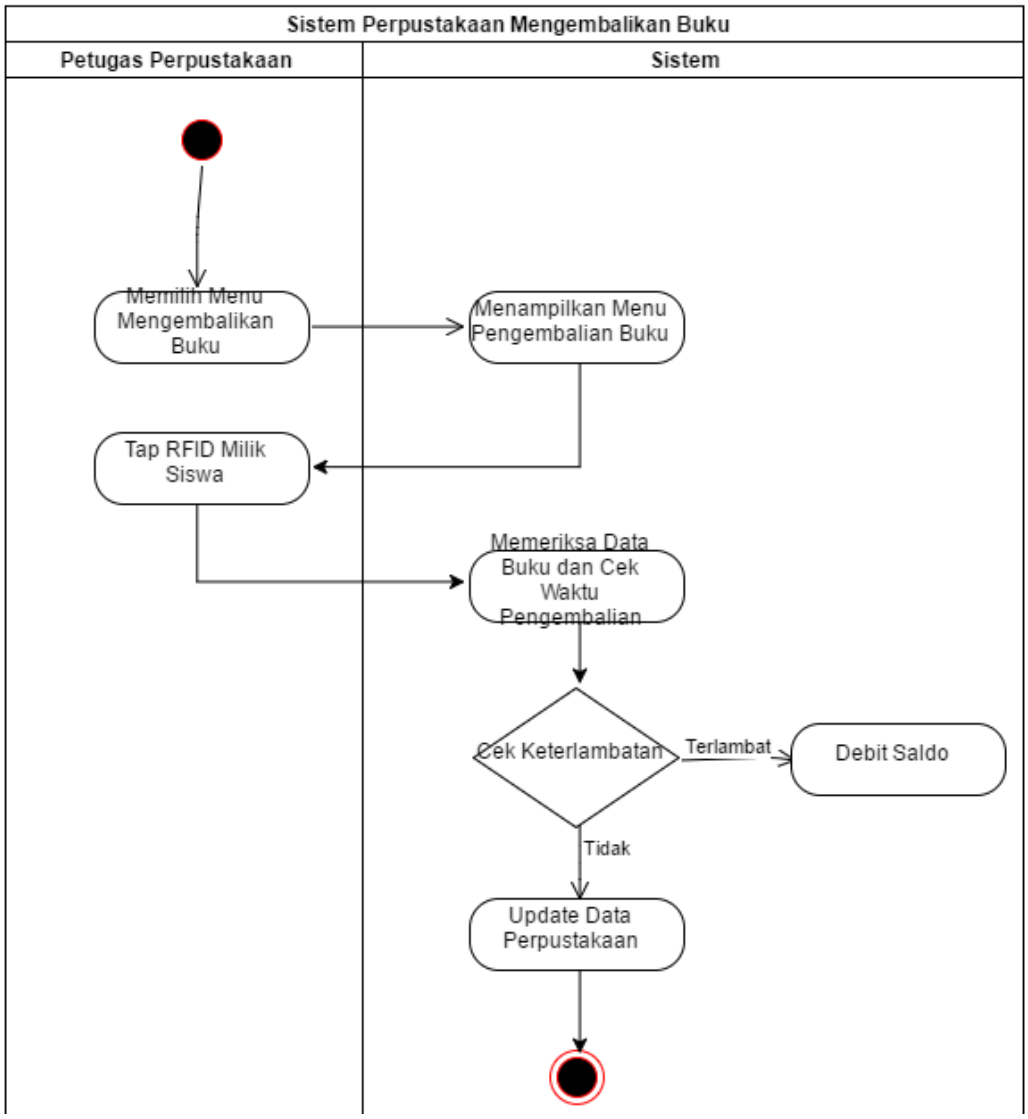
Gambar 3.27 Activity Diagram Rincian Parkir

n. *Activity* Diagram Peminjaman Buku



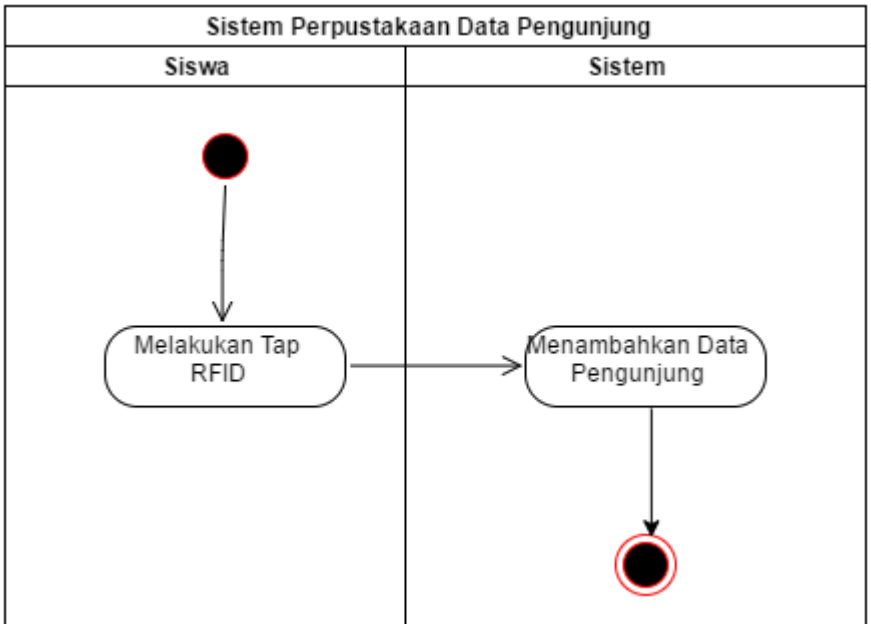
Gambar 3.28 Activity Diagram Peminjaman Buku

o. *Activity* Diagram Pengembalian Buku



Gambar 3.29 Activity Diagram Pengembalian Buku

p. *Activity* Diagram Data Pengunjung

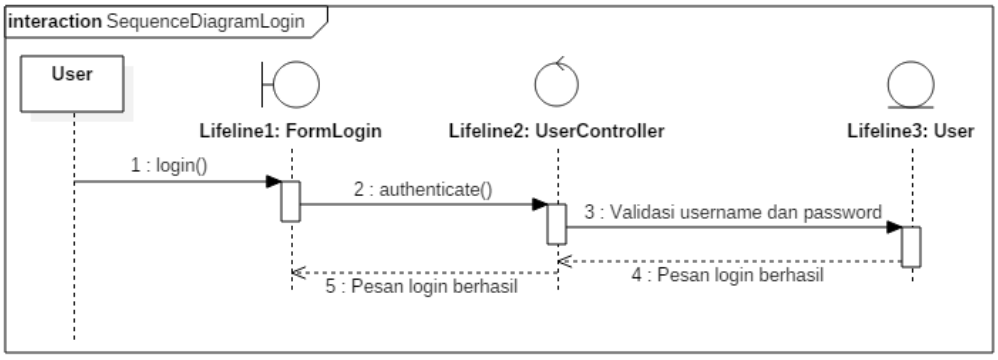


Gambar 3.30 Activity Diagram Kelola Data Pengunjung Perpustakaan

3.7.3 Sequence Diagram

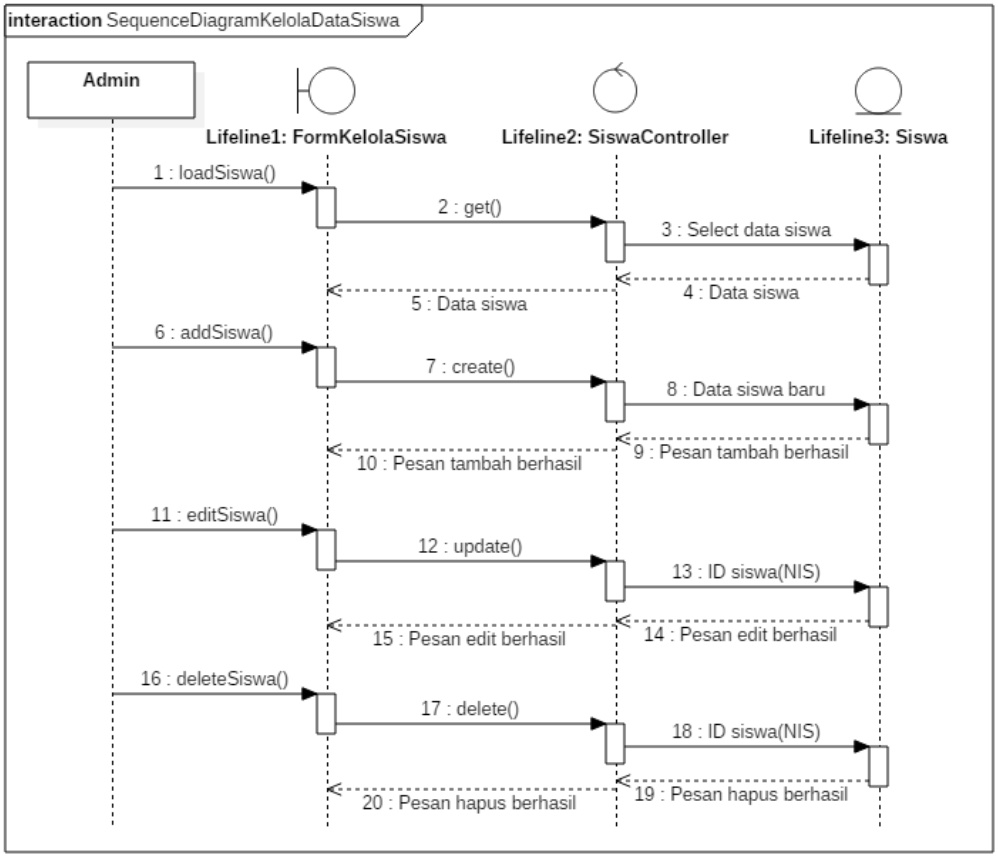
Sequence Diagram, menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek.

a. Sequence Diagram Login



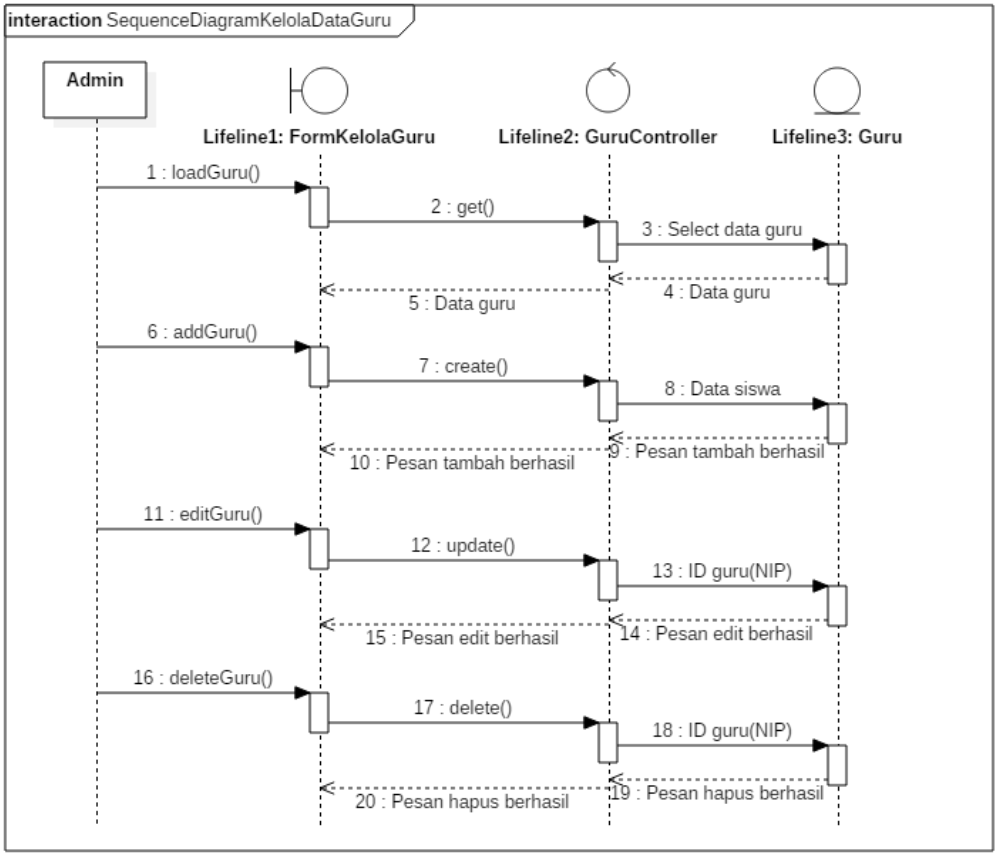
Gambar 3.31 Sequence Diagram Login

b. Sequence Diagram Kelola Data Siswa



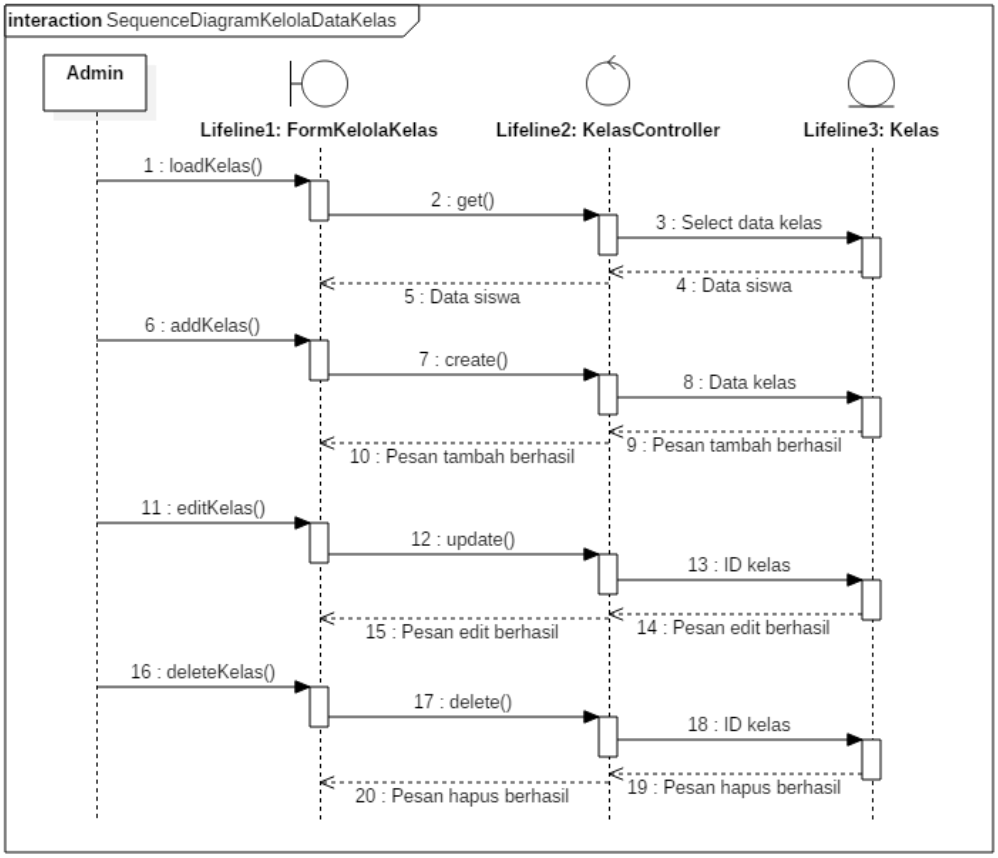
Gambar 3.32 Sequence Diagram Kelola Data Siswa

c. Sequence Diagram Kelola Data Guru



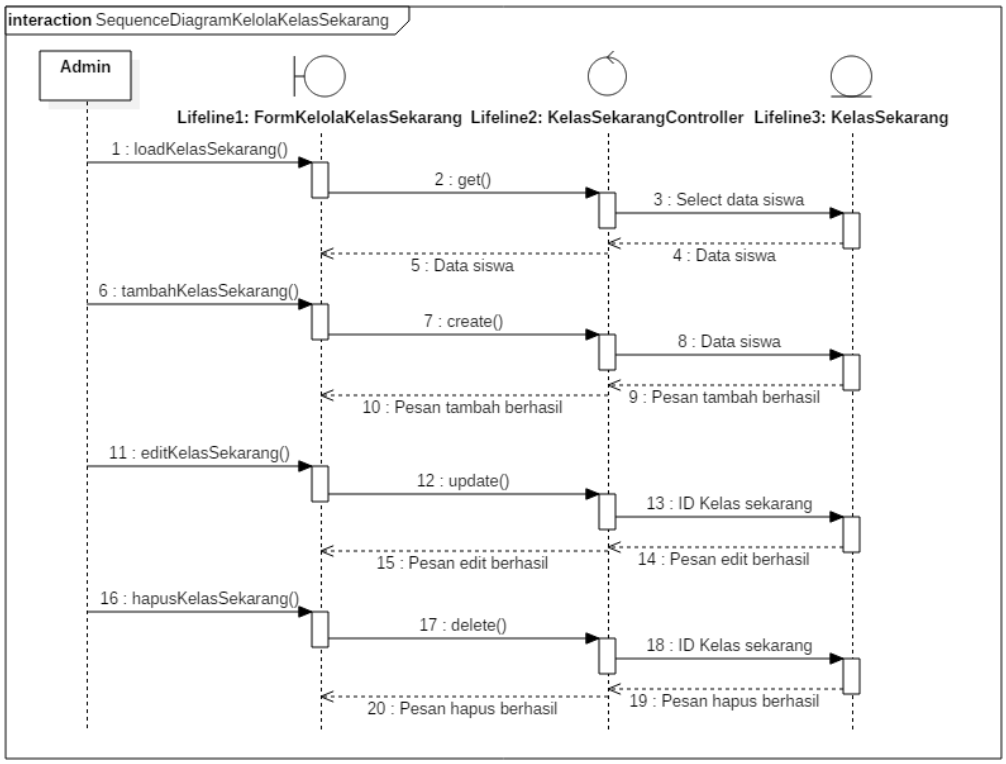
Gambar 3.33 Sequence Diagram Kelola Data Guru

d. Sequence Diagram Kelola Data Kelas



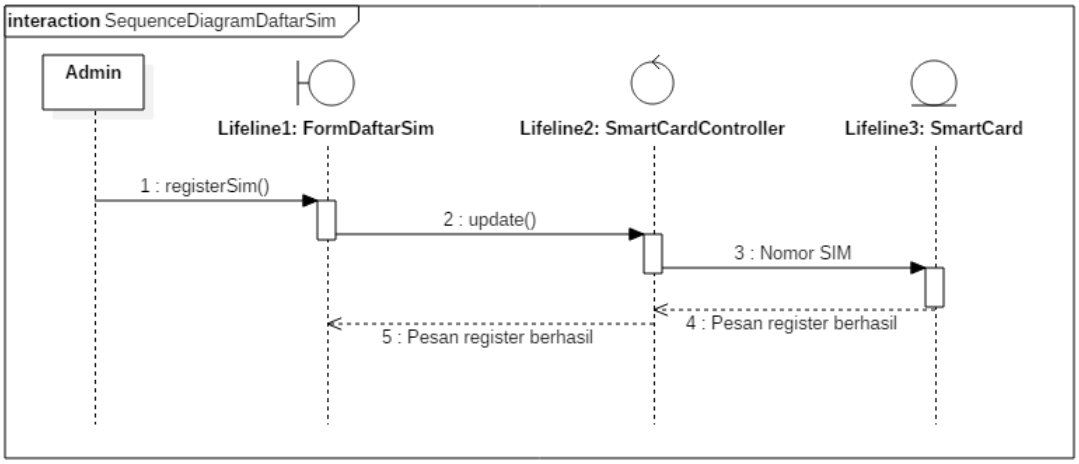
Gambar 3.34 Sequence Diagram Kelola Data Kelas

e. Sequence Diagram Write Kelola Data Kelas Sekarang



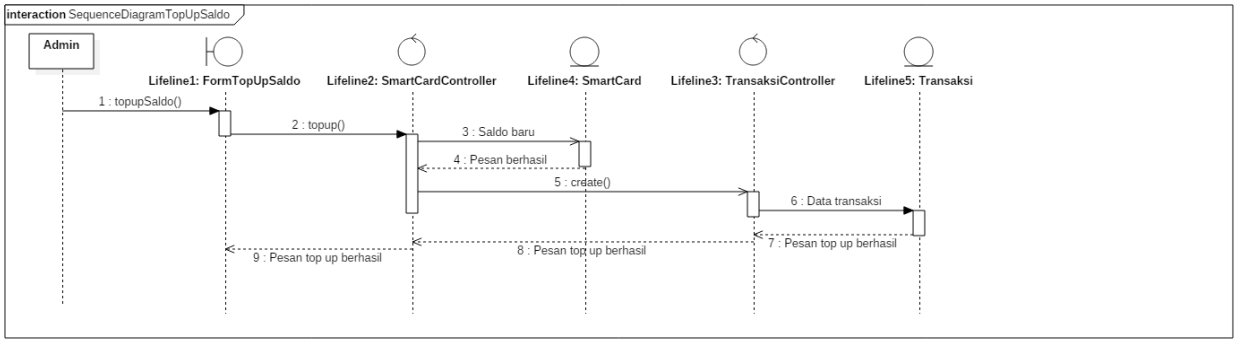
Gambar 3.35 Sequence Diagram Kelola Data Kelas Sekarang

f. Sequence Diagram Daftar SIM



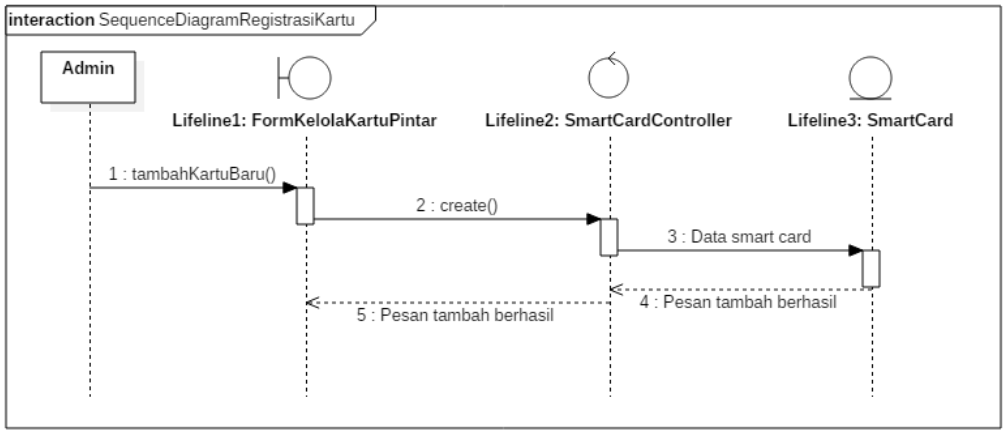
Gambar 3.36 Sequence Diagram Daftar SIM

g. Sequence Diagram Top Up Saldo



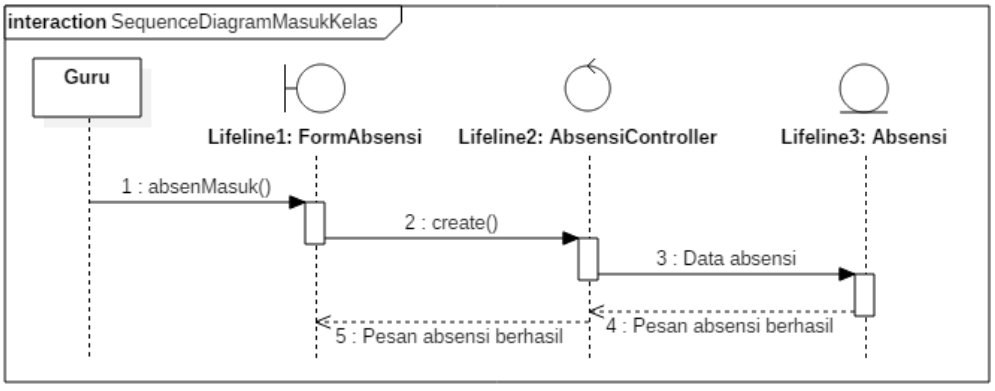
Gambar 3.37 Sequence Diagram Top Up Saldo

h. Sequence Diagram Registrasi Kartu



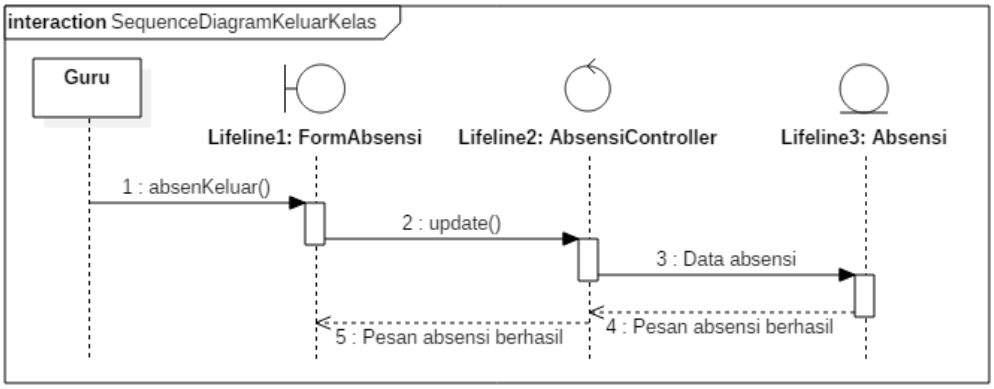
Gambar 3.38 Sequence Diagram Registrasi Kartu

i. Sequence Diagram Absen Datang



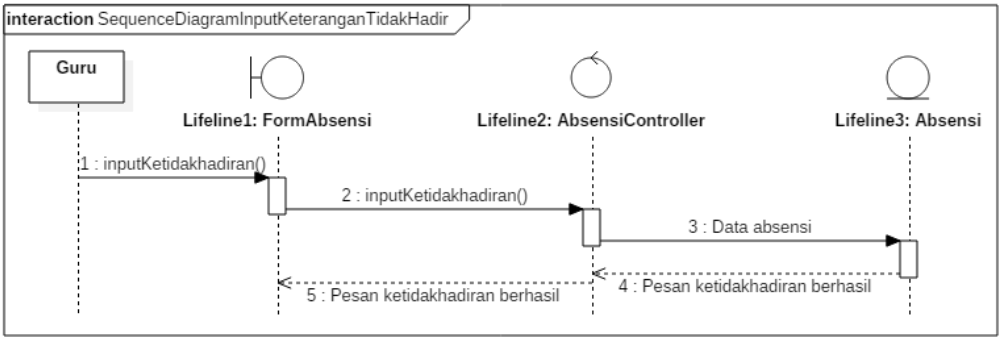
Gambar 3.39 Sequence Diagram Absen Datang

j. Sequence Diagram Absen Pulang



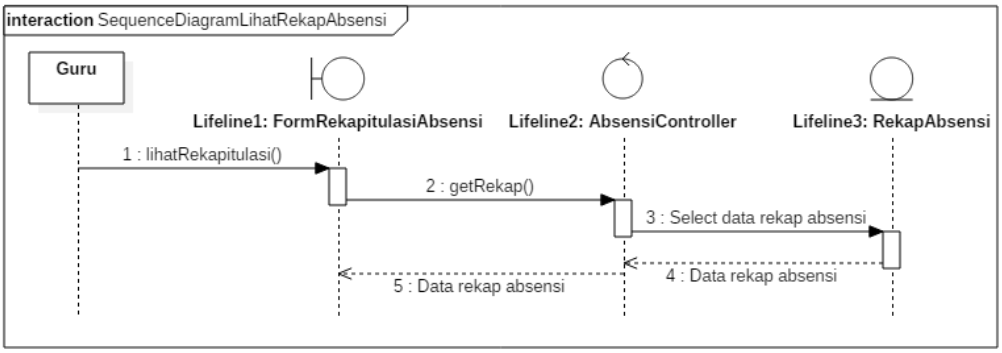
Gambar 3.40 Sequence Diagram Absen Pulang

k. Sequence Diagram Input Keterangan



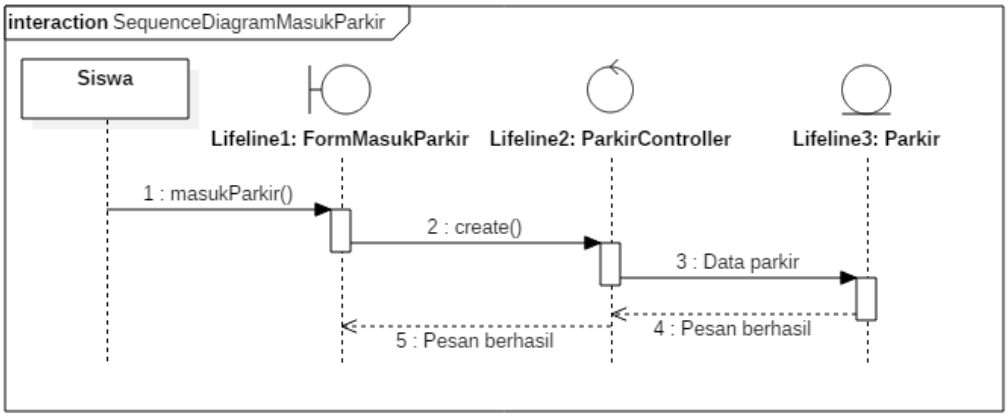
Gambar 3.41 Sequence Diagram Input Keterangan

l. Sequence Diagram Rekapitulasi Absensi



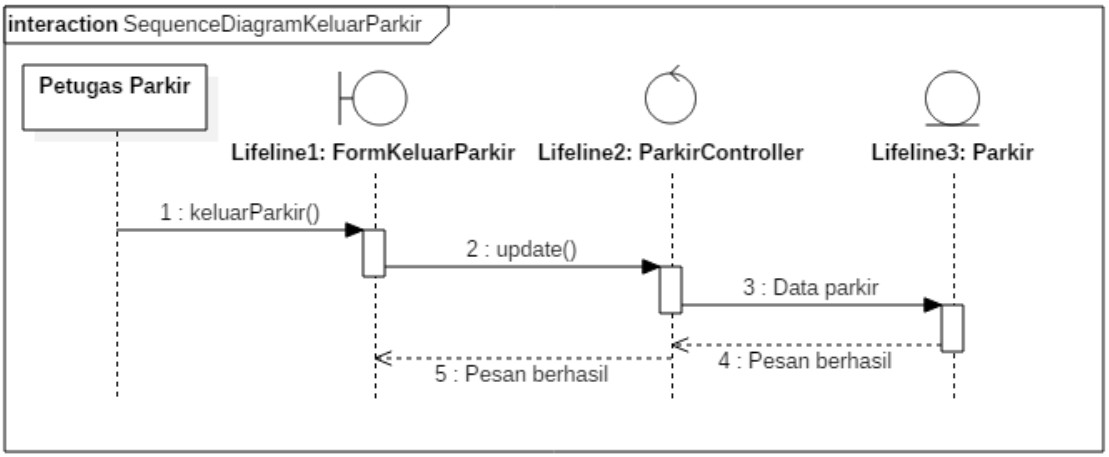
Gambar 3.42 Sequence Diagram Rekapitulasi Absensi

m. Sequence Diagram Masuk Parkir



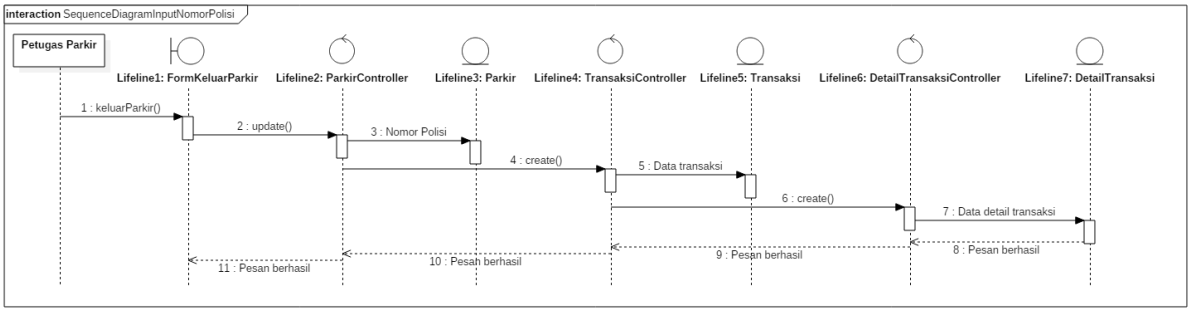
Gambar 3.43 Sequence Diagram Masuk Parkir

n. Sequence Diagram Keluar Parkir



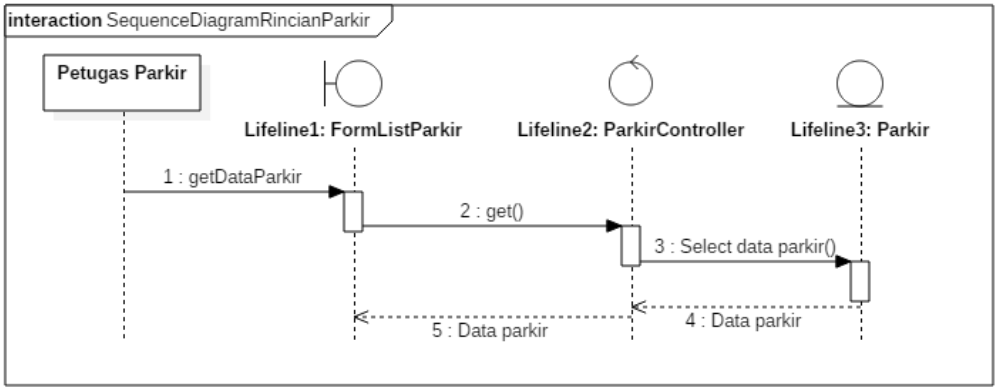
Gambar 3.44 Sequence Diagram Keluar Parkir

o. Sequence Diagram Input Nomor Polisi



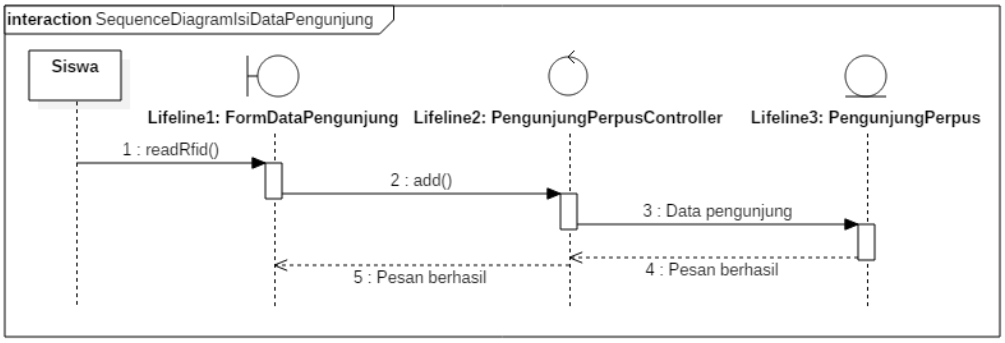
Gambar 3.45 Sequence Diagram Input Nomor Polisi

p. Sequence Diagram Rincian Parkir



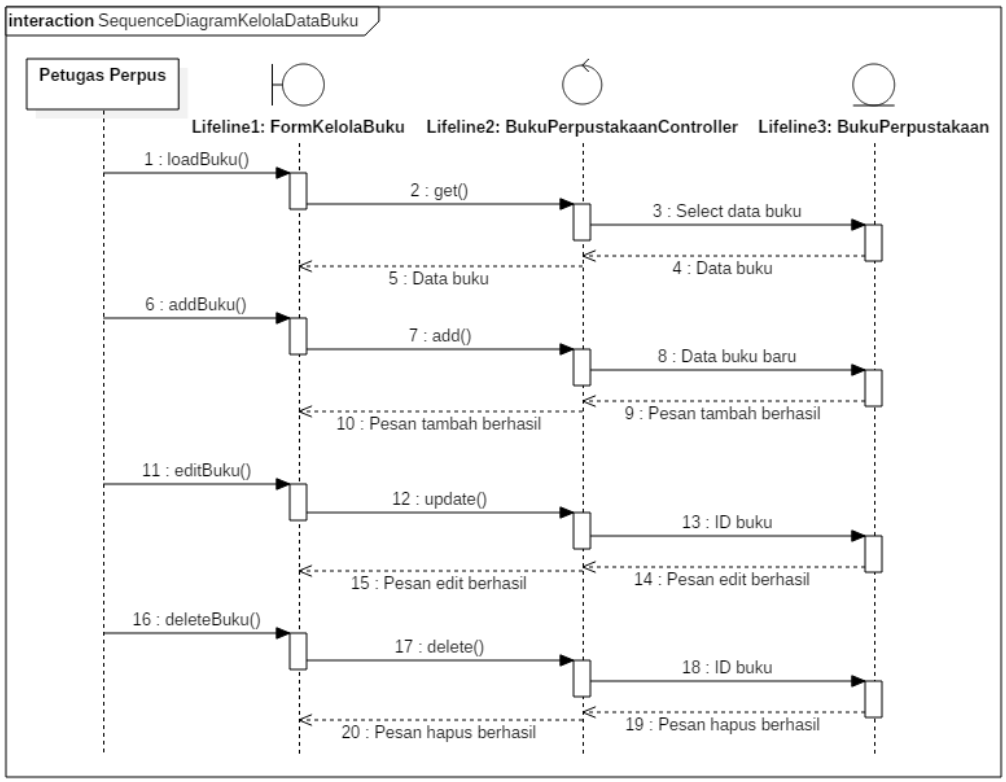
Gambar 3.46 Sequence Diagram Rincian Parkir

q. Sequence Diagram Data Pengunjung



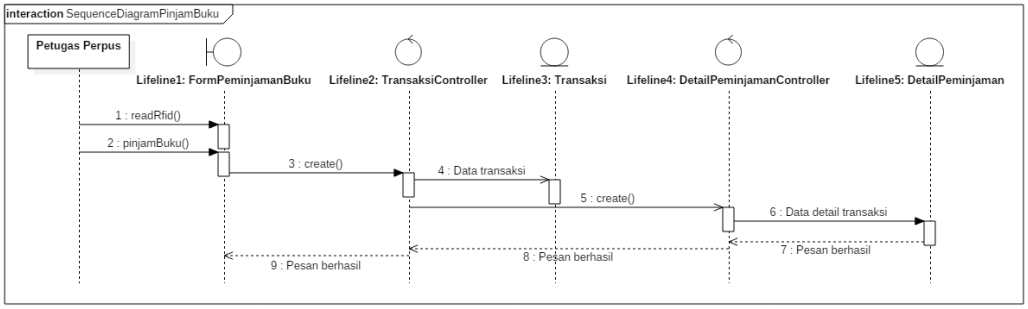
Gambar 3.47 Sequence Diagram Data Pengunjung

r. Sequence Diagram Kelola Data Buku



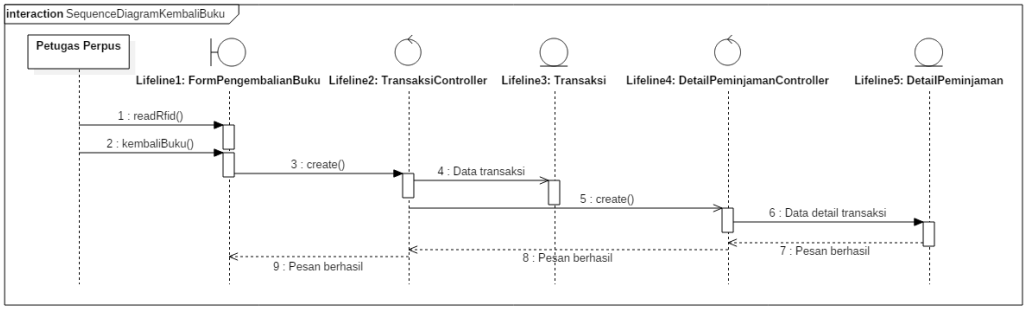
Gambar 3.48 Sequence Diagram Kelola Data Buku

s. Sequence Diagram Peminjaman Buku



Gambar 3.49 Sequence Diagram Peminjaman Buku

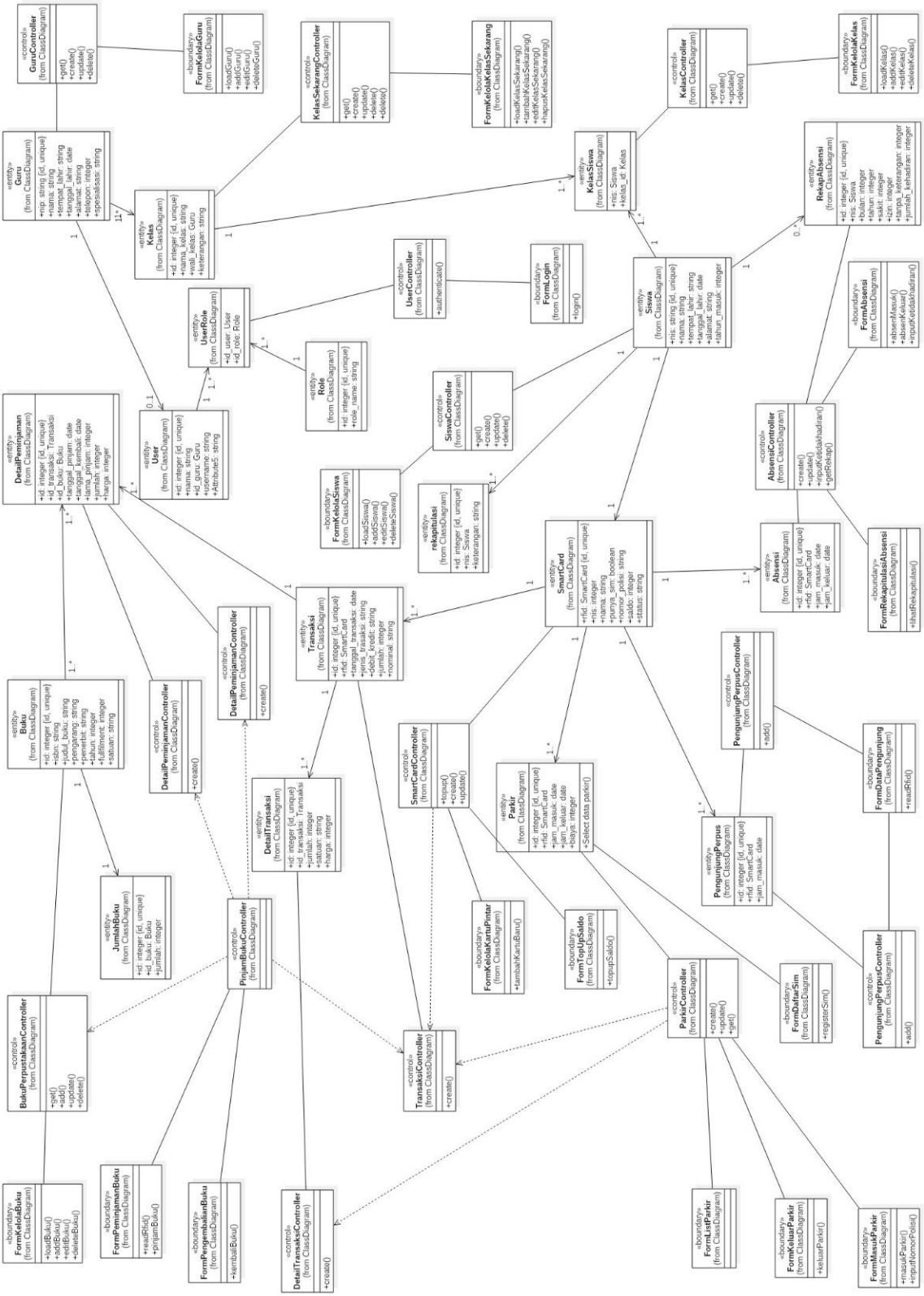
t. Sequence Diagram Pengembalian Buku



Gambar 3.50 Sequence Diagram Pengembalian Buku

3.7.4 Class Diagram

Terdapat tiga jenis kelas dalam sistem, yaitu kelas *Boundary*, kelas *Control*, dan kelas *Entity*. Kelas *Boundary* merupakan kelas yang digunakan untuk membangun tampilan antarmuka sistem. Kelas *Control* merupakan kelas yang digunakan sebagai tempat terjadinya seluruh proses yang ada dalam sistem dan juga validator yang memeriksa validasi data yang masuk, dan kelas *Entity* merupakan kelas yang digunakan sebagai penampung data atau representasi data itu sendiri. Seluruh kelas-kelas yang ada dalam sistem dapat dilihat pada Gambar .



Gambar 3.51 Class Diagram

Setiap kelas pada *Class* Diagram Gambar 3.55 memiliki jenis kelasnya masing-masing. Untuk lebih lengkapnya seperti yang terlihat pada Tabel 3.25

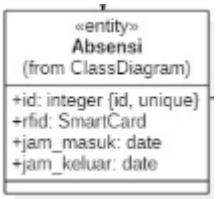
Tabel 3.25 Jenis Kelas setiap Class Diagram

No	Nama Kelas	Jenis Kelas
1	Absensi	Entity
2	Buku	Entity
3	DetailPeminjaman	Entity
4	DetailTransaksi	Entity
5	Parkir	Entity
6	RekapitulasiAbsensi	Entity
7	SmartCard	Entity
8	Transaksi	Entity
9	Kelas	Entity
10	KelasSiswa	Entity
11	Guru	Entity
12	Siswa	Entity
13	UserRole	Entity
14	PengunjungPerpus	Entity
15	Ketidakhadiran	Entity
16	User	Entity
17	Role	Entity
18	JumlahBuku	Entity
19	SiswaController	Controller
20	GuruController	Controller
21	KelasController	Controller
22	KelasSiswaController	Controller
23	AbsensiController	Controller
24	BukuController	Controller
25	TransaksiController	Controller
26	SmartCardController	Controller
27	Parkir	Controller

28	DetailTransaksi	Controller
29	DetailPeminjamanController	Controller
30	PinjamBukuController	Controller
31	PengunjungPerpusController	Controller
32	FormKelolaSiswa	Boundary
33	FormKelolaKelas	Boundary
34	FormKelolaKelasSiswa	Boundary
35	FormTopUpSaldo	Boundary
36	FormKelolaKartuPintar	Boundary
37	FormAbsensi	Boundary
38	FormRekapitulasiAbsensi	Boundary
39	FormMasukParkir	Boundary
40	FormKeluarParkir	Boundary
41	FormListParkir	Boundary
42	FormDataPengunjung	Boundary
43	FormPeminjamanBuku	Boundary
44	FormPengembalianBuku	Boundary
45	FormKelolaBuku	Boundary
46	FormLogin	Boundary
47	FormKelolaGuru	Boundary
48	FormDaftarSIM	Boundary

3.7.4.1 Kelas Absensi

Kelas Absensi merupakan kelas *Entity* yang berisi atribut yang berhubungan dengan data Absensi. Gambar 3.58 merupakan gambar dari kelas Absensi.

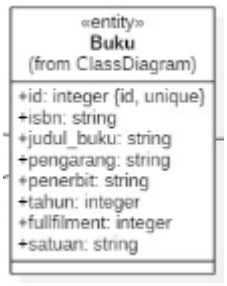


Gambar 3.52 Kelas Absensi

Pada kelas Absensi hanya terdapat atribut karena kelas Absensi merupakan kelas *Entity* dan semua operasi telah dilakukan oleh kelas AbsensiController. Pada Tabel 3.26 adalah deskripsi dari atribut yang ada pada kelas Absensi.

3.7.4.2 Kelas Buku

Kelas Absensi merupakan kelas *Entity* yang berisi atribut yang berhubungan dengan data Absensi. Gambar 3.59 merupakan gambar dari kelas Absensi.



Gambar 3.53 Kelas Buku

Pada kelas Absensi hanya terdapat atribut karena kelas Absensi merupakan kelas *Entity* dan semua operasi telah dilakukan oleh kelas AbsensiController. Pada Tabel 3.27 adalah deskripsi dari atribut yang ada pada kelas Buku.

3.7.4.3 Kelas Detail Peminjaman

Kelas Absensi merupakan kelas *Entity* yang berisi atribut yang berhubungan dengan data Absensi. Gambar 3.60 merupakan gambar dari kelas Absensi.



Gambar 3.54 Kelas Detail Peminjaman

Pada kelas Absensi hanya terdapat atribut karena kelas Absensi merupakan kelas *Entity* dan semua operasi telah dilakukan oleh kelas AbsensiController. Pada Tabel 3.28 adalah deskripsi dari atribut yang ada pada kelas Pesanan.

3.7.4.4 Kelas Detail Transaksi

Kelas Absensi merupakan kelas *Entity* yang berisi atribut yang berhubungan dengan data Absensi. Gambar 3.61 merupakan gambar dari kelas Absensi.

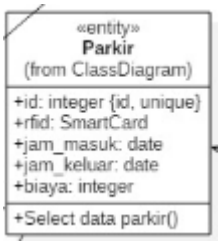


Gambar 3.55 Kelas Detail Transaksi

Pada kelas Absensi hanya terdapat atribut karena kelas Absensi merupakan kelas *Entity* dan semua operasi telah dilakukan oleh kelas AbsensiController. Pada Tabel 3.29 adalah deskripsi dari atribut yang ada pada kelas Pesanan.

3.7.4.5 Kelas Parkir

Kelas Parkir merupakan kelas *Entity* yang berisi atribut yang berhubungan dengan data Parkir. Gambar 3.41 merupakan gambar dari kelas Parkir.



Gambar 3.56 Kelas Parkir

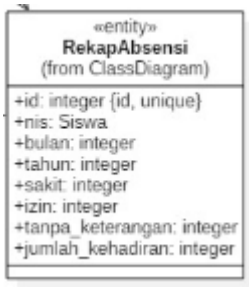
Pada kelas Parkir hanya terdapat atribut karena kelas Parkir merupakan kelas *Entity* dan semua operasi telah dilakukan oleh kelas Parkir Controller. Pada Tabel 3.30 adalah deskripsi dari atribut yang ada pada kelas Parkir.

Tabel 3.26 Atribut kelas parkir

Nama	Tipe
rfid	SmartCard
Jam_masuk	Date
Jam_keluar	Date
biaya	integer

3.7.4.6 Kelas Rekapitulasi Absensi

Kelas Rekapitulasi Absensi merupakan kelas *Entity* yang berisi atribut yang berhubungan dengan data Rekapitulasi Absensi. Gambar 3.41 merupakan gambar dari kelas Absensi.



Gambar 3.57 Kelas Rekapitulasi Absensi

Pada kelas Rekapitulasi Absensi hanya terdapat atribut karena kelas Absensi merupakan kelas *Entity* dan semua operasi telah dilakukan oleh kelas AbsensiController.

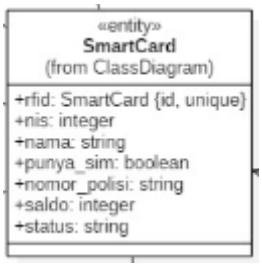
Pada Tabel 3.31 adalah deskripsi dari atribut yang ada pada kelas Rekapitulasi Absensi.

Tabel 3.27 Atribut Kelas Rekapitulasi Absensi

Nama	Tipe
nis	String
Bulan	integer
Tahun	integer
sakit	integer
izin	integer
Tanpa_keterangan	integer
Jumlah_kehadiran	integer

3.7.4.7 Kelas Smart Card

Kelas Smart Card merupakan kelas *Entity* yang berisi atribut yang berhubungan dengan data Smart Card. Gambar 3.41 merupakan gambar dari kelas Smart Card.



Gambar 3.58 Kelas Smart Card

Pada kelas Smart Card hanya terdapat atribut karena kelas Smart Card merupakan kelas *Entity* dan semua operasi telah dilakukan oleh kelas SmartCardController.

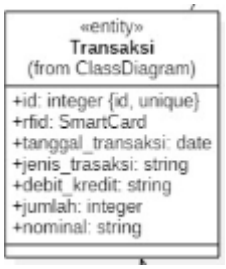
Pada Tabel 3.32 adalah deskripsi dari atribut yang ada pada kelas Smart Card.

Tabel 3.28 Atribut kelas SmartCard

Nama	Tipe
nis	String
Nama	String
Punya_sim	boolean
No_polis	String
Saldo	integer
status	String

3.7.4.8 Kelas Transaksi

Kelas Transaksi merupakan kelas *Entity* yang berisi atribut yang berhubungan dengan data Transaksi. Gambar 3.41 merupakan gambar dari kelas Transaksi.



Gambar 3.59 Kelas Transaksi

Pada kelas Transaksi hanya terdapat atribut karena kelas Transaksi merupakan kelas *Entity* dan semua operasi telah dilakukan oleh kelas TransaksiController.

Pada Tabel 3.33 adalah deskripsi dari atribut yang ada pada kelas Transaksi.

Tabel 3.29 Atribut Kelas Transaksi

Nama	Tipe
id	integer
rfid	string
Tanggal_transaksi	date
Jenis_transaksi	string
Debet_kredit	string
jumlah	integer
nominal	string

3.7.4.9 Kelas Kelas

Kelas Kelas merupakan kelas *Entity* yang berisi atribut yang berhubungan dengan data Absensi. Gambar 3.41 merupakan gambar dari kelas Kelas.



Gambar 3.60 Kelas Kelas

Pada kelas Kelas hanya terdapat atribut karena kelas Absensi merupakan kelas *Entity* dan semua operasi telah dilakukan oleh kelas Kelas Controller.

Pada Tabel 3.34 adalah deskripsi dari atribut yang ada pada kelas Kelas.

Tabel 3.30 Atribut kelas Kelas

Nama	Tipe
id	integer

Nama_kelas	string
Wakul_kelas	string
keterangan	string

3.7.4.10 Kelas Kelas Siswa

Kelas Kelas Siswa merupakan kelas *Entity* yang berisi atribut yang berhubungan dengan data Kelas Siswa. Gambar 3.41 merupakan gambar dari kelas Kelas Siswa.



Gambar 3.61 Kelas Kelas Siswa

Pada kelas Kelas Siswa hanya terdapat atribut karena kelas Kelas Siswa merupakan kelas *Entity* dan semua operasi telah dilakukan oleh kelas KelasSiswaController.

Pada Tabel 3.35 adalah deskripsi dari atribut yang ada pada kelas Kelas Siswa.

Tabel 3.31 Atribut kelas Kelas Siswa

Nama	Tipe
Nis	string
Kelas_id	integer

3.7.4.11 Kelas Guru

Kelas Guru merupakan kelas *Entity* yang berisi atribut yang berhubungan dengan data Guru. Gambar 3.41 merupakan gambar dari kelas Guru.



Gambar 3.62 Kelas Guru

Pada kelas Guru hanya terdapat atribut karena kelas Guru merupakan kelas *Entity* dan semua operasi telah dilakukan oleh kelas GuruController.

Pada Tabel 3.36 adalah deskripsi dari atribut yang ada pada kelas Guru.

Tabel 3.32 Atribut Kelas Guru

Nama	Tipe
nip	string
nama	string
Tempal_lahir	string
Tanggal_lahir	date
alamat	string
telepon	integer
Spesialis	string

3.7.4.12 Kelas Siswa

Kelas Siswa merupakan kelas *Entity* yang berisi atribut yang berhubungan dengan data Siswa. Gambar 3.41 merupakan gambar dari kelas Siswa.

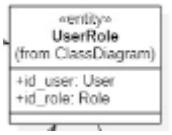


Gambar 3.63 Kelas Siswa

Pada kelas Siswa hanya terdapat atribut karena kelas Siswa merupakan kelas *Entity* dan semua operasi telah dilakukan oleh kelas SiswaController.

3.7.3.13 Kelas User Role

Kelas User Role merupakan kelas *Entity* yang berisi atribut yang berhubungan dengan data User Role. Gambar 3.41 merupakan gambar dari kelas User Role.

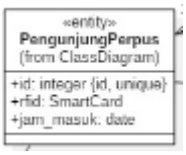


Gambar 3.64 Kelas User Role

Pada kelas User Role hanya terdapat atribut karena kelas User Role merupakan kelas *Entity* dan semua operasi telah dilakukan oleh kelas UserRoleController.

3.7.4.13 Kelas Pengunjung Perpus

Kelas Pengunjung Perpus merupakan kelas *Entity* yang berisi atribut yang berhubungan dengan data Pengunjung Perpus. Gambar 3.41 merupakan gambar dari kelas Pengunjung Perpus.



Gambar 3.65 Kelas Pengunjung Perpus

Pada kelas Pengunjung Perpus hanya terdapat atribut karena kelas Pengunjung Perpus merupakan kelas *Entity* dan semua operasi telah dilakukan oleh kelas PengunjungPerpusController.

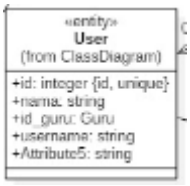
3.7.4.14 Kelas Ketidak Hadiran

Kelas Ketidak Hadiran merupakan kelas *Entity* yang berisi atribut yang berhubungan dengan data Ketidak Hadiran. Gambar 3.41 merupakan gambar dari kelas Ketidak Hadiran.

Pada kelas Ketidak Hadiran hanya terdapat atribut karena kelas Ketidak Hadiran merupakan kelas *Entity* dan semua operasi telah dilakukan oleh kelas KetidakHadiranController.

3.7.4.15 Kelas User

Kelas User merupakan kelas *Entity* yang berisi atribut yang berhubungan dengan data User. Gambar 3.72 merupakan gambar dari kelas User.

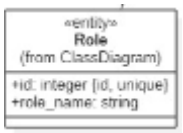


Gambar 3.66 Kelas User

Pada kelas User hanya terdapat atribut karena kelas Absensi merupakan kelas *Entity* dan semua operasi telah dilakukan oleh kelas UserController.

3.7.4.16 Kelas Role

Kelas Role merupakan kelas *Entity* yang berisi atribut yang berhubungan dengan data Role. Gambar 3.41 merupakan gambar dari kelas Role.



Gambar 3.67 Kelas Role

Pada kelas Role hanya terdapat atribut karena kelas Absensi merupakan kelas *Entity* dan semua operasi telah dilakukan oleh kelas Role Controller.

3.7.4.17 Kelas Jumlah Buku

Kelas Jumlah Buku merupakan kelas *Entity* yang berisi atribut yang berhubungan dengan data Jumlah Buku. Gambar 3.41 merupakan gambar dari kelas Jumlah Buku.



Gambar 3.68 Kelas Jumlah Buku

Pada kelas Jumlah Buku hanya terdapat atribut karena kelas Jumlah Buku merupakan kelas *Entity* dan semua operasi telah dilakukan oleh kelas JumlahBukuController.

3.7.4.18 Kelas SiswaController

SiswaController merupakan kelas yang melakukan operasi dari 1 kelas *Boundary*, yaitu FormKelolaSiswa dan 1 kelas entity yaitu Siswa. Gambar 3.75 merupakan gambar dari kelas ControlPemesanan.



Gambar 3.69 Kelas SiswaController

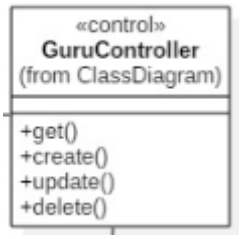
Gambar 3.1 Kelas ControlPemesanan

Pada kelasi SiswaController terdapat 4 operasi yang menghubungkan kelas FormKelolaSiswa dan 1 kelas entity yaitu Siswa. Berikut deskripsi dari operasi yang ada pada kelas ControlPemesanan.

1) *getData(),*
2) *create(),*
3) *update(),*
4) *delete(),*

3.7.4.19 Kelas GuruController

GuruController merupakan kelas yang melakukan operasi dari 3 kelas *Boundary*, yaitu FormPesanan, FormPemesanan dan FormPengguna. Gambar 3.29 merupakan gambar dari kelas ControlPemesanan.



Gambar 3.70 Kelas GuruController

Pada kelasi ControlPemesanan terdapat 5 operasi yang menghubungkan kelas FormPesanan, FormPemesanan dan FormPengguna dengan kelas *Entity* Pesanan dan Tiket. Berikut deskripsi dari operasi yang ada pada kelas ControlPemesanan.

1) *getData(),*
2) *create(),*
3) *update(),*
4) *delete(),*

3.7.4.20 Kelas KelasController

ControlPemesanan merupakan kelas yang melakukan operasi dari 3 kelas *Boundary*, yaitu FormPesanan, FormPemesanan dan FormPengguna. Gambar 3.29 merupakan gambar dari kelas ControlPemesanan.

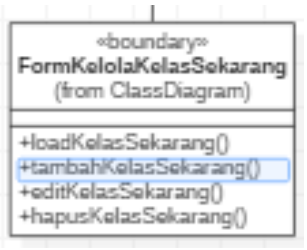


Gambar 3.71 KelasController

Pada kelas ControlPemesanan terdapat 5 operasi yang menghubungkan kelas FormPesanan, FormPemesanan dan FormPengguna dengan kelas *Entity* Pesanan dan Tiket. Berikut deskripsi dari operasi yang ada pada kelas ControlPemesanan.

1) *getData(),*
2) *create(),*
3) *update(),*
4) *delete(),*

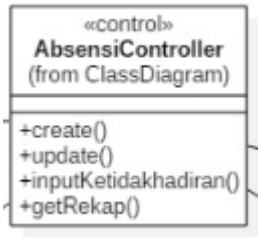
3.7.4.21 Kelas KelasSiswaController



Gambar 3.72 Kelas KelasSiswaController

3.7.4.22 Kelas AbsensiController

ControlPemesanan merupakan kelas yang melakukan operasi dari 3 kelas *Boundary*, yaitu FormPesanan, FormPemesanan dan FormPengguna. Gambar 3.80 merupakan gambar dari kelas ControlPemesanan.



Gambar 3.73 Kelas AbsensiController

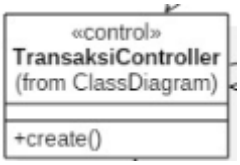
Pada kelasi ControlPemesanan terdapat 5 operasi yang menghubungkan kelas FormPesanan, FormPemesanan dan FormPengguna dengan kelas *Entity* Pesanan dan Tiket. Berikut deskripsi dari operasi yang ada pada kelas ControlPemesanan.

1) *getData(),*

2) *create(),*
3) *update(),*
4) *delete(),*

3.7.4.23 Kelas TransaksiController

ControlPemesanan merupakan kelas yang melakukan operasi dari 3 kelas *Boundary*, yaitu FormPesanan, FormPemesanan dan FormPengguna. Gambar 3.29 merupakan gambar dari kelas ControlPemesanan.



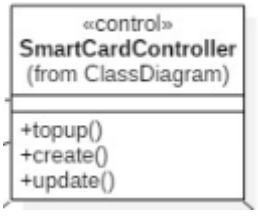
Gambar 3.74 Kelas TransaksiController

Pada kelasi ControlPemesanan terdapat 5 operasi yang menghubungkan kelas FormPesanan, FormPemesanan dan FormPengguna dengan kelas *Entity* Pesanan dan Tiket. Berikut deskripsi dari operasi yang ada pada kelas ControlPemesanan.

1) *getData(),*

3.7.4.24 Kelas SmartCardController

ControlPemesanan merupakan kelas yang melakukan operasi dari 3 kelas *Boundary*, yaitu FormPesanan, FormPemesanan dan FormPengguna. Gambar 3.83 merupakan gambar dari kelas ControlPemesanan.



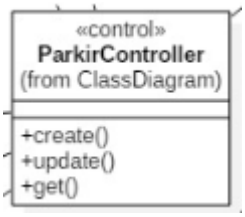
Gambar 3.75 Kelas SmartCardController

Pada kelasi ControlPemesanan terdapat 5 operasi yang menghubungkan kelas FormPesanan, FormPemesanan dan FormPengguna dengan kelas *Entity* Pesanan dan Tiket. Berikut deskripsi dari operasi yang ada pada kelas ControlPemesanan.

1) *topup(),*
2) *create(),*
3) *update(),*

3.7.4.25 Kelas ParkirController

ControlPemesanan merupakan kelas yang melakukan operasi dari 3 kelas *Boundary*, yaitu FormPesanan, FormPemesanan dan FormPengguna. Gambar 3.84 merupakan gambar dari kelas ControlPemesanan.



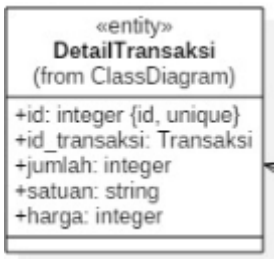
Gambar 3.76 Kelas Parkir Controller

Pada kelasi ControlPemesanan terdapat 5 operasi yang menghubungkan kelas FormPesanan, FormPemesanan dan FormPengguna dengan kelas *Entity* Pesanan dan Tiket. Berikut deskripsi dari operasi yang ada pada kelas ControlPemesanan.

1) *create(),*
2) *update(),*
3) *get(),*

3.7.4.26 Kelas DetailController

ControlPemesanan merupakan kelas yang melakukan operasi dari 3 kelas *Boundary*, yaitu FormPesanan, FormPemesanan dan FormPengguna. Gambar 3.85 merupakan gambar dari kelas ControlPemesanan.



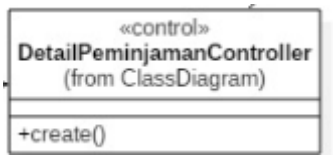
Gambar 3.77 Kelas DetailController

Pada kelasi ControlPemesanan terdapat 5 operasi yang menghubungkan kelas FormPesanan, FormPemesanan dan FormPengguna dengan kelas *Entity* Pesanan dan Tiket. Berikut deskripsi dari operasi yang ada pada kelas ControlPemesanan.

1) *getData(),*
2) *create(),*
3) *update(),*
4) *delete(),*

3.7.4.27 Kelas DetailPeminjamanController

ControlPemesanan merupakan kelas yang melakukan operasi dari 3 kelas *Boundary*, yaitu FormPesanan, FormPemesanan dan FormPengguna. Gambar 3.86 merupakan gambar dari kelas ControlPemesanan.



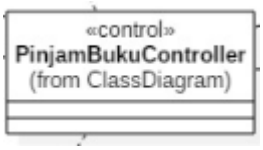
Gambar 3.78 Kelas DetailPeminjamanController

Pada kelasi ControlPemesanan terdapat 5 operasi yang menghubungkan kelas FormPesanan, FormPemesanan dan FormPengguna dengan kelas *Entity* Pesanan dan Tiket. Berikut deskripsi dari operasi yang ada pada kelas ControlPemesanan.

1) *create(),*

3.7.4.28 Kelas PinjamBukuController

ControlPemesanan merupakan kelas yang melakukan operasi dari 3 kelas *Boundary*, yaitu FormPesanan, FormPemesanan dan FormPengguna. Gambar 3.87 merupakan gambar dari kelas ControlPemesanan.

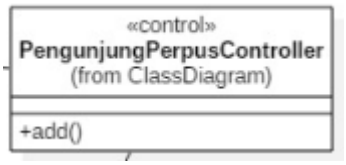


Gambar 3.79 Kelas PinjamanBukuController

Pada kelasi ControlPemesanan terdapat 5 operasi yang menghubungkan kelas FormPesanan, FormPemesanan dan FormPengguna dengan kelas *Entity* Pesanan dan Tiket. Berikut deskripsi dari operasi yang ada pada kelas ControlPemesanan.

3.7.4.29 Kelas PengunjungPerpusController

PengunjungPerpusController merupakan kelas yang melakukan operasi dari 3 kelas *Boundary*, yaitu FormPesanan, FormPemesanan dan FormPengguna. Gambar 3.88 merupakan gambar dari kelas ControlPemesanan.



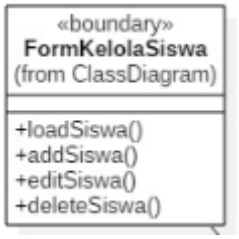
Gambar 3.80 Kelas PengunjungPerpusController

Pada kelasi ControlPemesanan terdapat 5 operasi yang menghubungkan kelas FormPesanan, FormPemesanan dan FormPengguna dengan kelas *Entity* Pesanan dan Tiket. Berikut deskripsi dari operasi yang ada pada kelas ControlPemesanan.

1) *add()*,

3.7.4.30 Kelas FormKelolaSiswa

FormKelolaSiswa merupakan antarmuka dari proses pemesanan yang dilakukan oleh pengunjung. FormPemesanan merupakan tempat interaksi pengunjung dengan sistem dalam proses pengisian data pemesanan tiket.

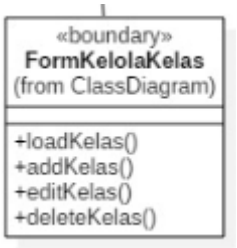


Gambar 3.81 Kelas FormKelolaSiswa

Pada FormKelolaSiswa terdapat satu operasi yang berfungsi untuk mengirimkan seluruh data pemesanan yaitu *submitPemesanan().* Operasi ini mengirimkan data pemesanan pada kelas ControlPemesanan untuk kemudian dilakukan *insert* pada *database*.

3.7.4.31 Kelas FormKelolaKelas

FormKelolaKelas merupakan antarmuka dari proses pemesanan yang dilakukan oleh pengunjung. FormPemesanan merupakan tempat interaksi pengunjung dengan sistem dalam proses pengisian data pemesanan tiket.

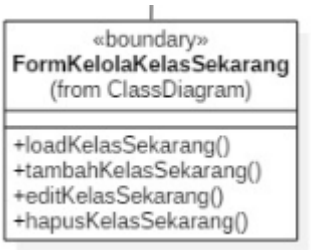


Gambar 3.82 Kelas FormKelolaKelas

Pada FormKelolaKelas terdapat satu operasi yang berfungsi untuk mengirimkan seluruh data pemesanan yaitu *submitPemesanan().* Operasi ini mengirimkan data pemesanan pada kelas ControlPemesanan untuk kemudian dilakukan *insert* pada *database*.

3.7.4.32 Kelas FormKelolaKelasSiswa

FormKelolaKelasSiswa merupakan antarmuka dari proses pemesanan yang dilakukan oleh pengunjung. FormPemesanan merupakan tempat interaksi pengunjung dengan sistem dalam proses pengisian data pemesanan tiket.

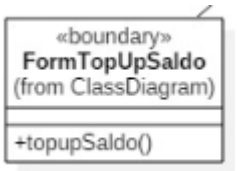


Gambar 3.83 Kelas FormKelolaKelasSiswa

Pada FormKelolaKelasSiswa terdapat satu operasi yang berfungsi untuk mengirimkan seluruh data pemesanan yaitu *submitPemesanan().* Operasi ini mengirimkan data pemesanan pada kelas ControlPemesanan untuk kemudian dilakukan *insert* pada *database*.

3.7.4.33 Kelas FormTopUpSaldo

FormTopUpSaldo merupakan antarmuka dari proses pemesanan yang dilakukan oleh pengunjung. FormPemesanan merupakan tempat interaksi pengunjung dengan sistem dalam proses pengisian data pemesanan tiket.



Gambar 3.84 Kelas FormTopUpSaldo

Pada FormTopUpSaldo terdapat satu operasi yang berfungsi untuk mengirimkan seluruh data pemesanan yaitu *submitPemesanan().* Operasi ini mengirimkan data pemesanan pada kelas ControlPemesanan untuk kemudian dilakukan *insert* pada *database*.

3.7.4.34 Kelas FormKelolaKartuPintar

FormKelolaKartuPintar merupakan antarmuka dari proses pemesanan yang dilakukan oleh pengunjung. FormPemesanan merupakan tempat interaksi pengunjung dengan sistem dalam proses pengisian data pemesanan tiket.

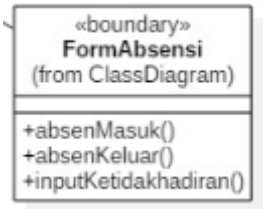


Gambar 3.85 Kelas FormKelolaKartuPintar

Pada FormKelolaKartuPintar terdapat satu operasi yang berfungsi untuk mengirimkan seluruh data pemesanan yaitu *submitPemesanan().* Operasi ini mengirimkan data pemesanan pada kelas ControlPemesanan untuk kemudian dilakukan *insert* pada *database*.

3.7.4.35 Kelas FormAbsensi

FormAbsensi merupakan antarmuka dari proses pemesanan yang dilakukan oleh pengunjung. FormPemesanan merupakan tempat interaksi pengunjung dengan sistem dalam proses pengisian data pemesanan tiket.

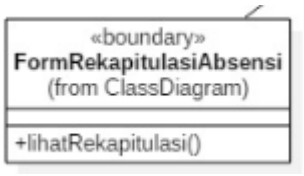


Gambar 3.86 Kelas FormAbsensi

Pada FormAbsensi terdapat satu operasi yang berfungsi untuk mengirimkan seluruh data pemesanan yaitu *submitPemesanan().* Operasi ini mengirimkan data pemesanan pada kelas ControlPemesanan untuk kemudian dilakukan *insert* pada *database*.

3.7.4.36 Kelas FormRekapitulasiAbsensi

FormRekapitulasiAbsensi merupakan antarmuka dari proses pemesanan yang dilakukan oleh pengunjung. FormPemesanan merupakan tempat interaksi pengunjung dengan sistem dalam proses pengisian data pemesanan tiket.

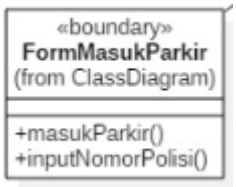


Gambar 3.87 Kelas Form Rekapitulasi Absensi

Pada FormRekapitulasiAbsensi terdapat satu operasi yang berfungsi untuk mengirimkan seluruh data pemesanan yaitu *submitPemesanan().* Operasi ini mengirimkan data pemesanan pada kelas ControlPemesanan untuk kemudian dilakukan *insert* pada *database*.

3.7.4.37 Kelas FormMasukParkir

FormMasukParkir merupakan antarmuka dari proses pemesanan yang dilakukan oleh pengunjung. FormPemesanan merupakan tempat interaksi pengunjung dengan sistem dalam proses pengisian data pemesanan tiket.

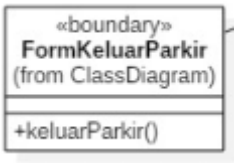


Gambar 3.88 Kelas FormMasukParkir

Pada FormMasukParkir terdapat satu operasi yang berfungsi untuk mengirimkan seluruh data pemesanan yaitu *submitPemesanan().* Operasi ini mengirimkan data pemesanan pada kelas ControlPemesanan untuk kemudian dilakukan *insert* pada *database*.

3.7.4.38 Kelas FormKeluarParkir

FormKeluarParkir merupakan antarmuka dari proses pemesanan yang dilakukan oleh pengunjung. FormPemesanan merupakan tempat interaksi pengunjung dengan sistem dalam proses pengisian data pemesanan tiket.

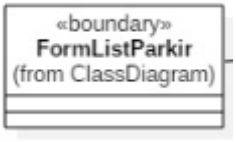


Gambar 3.89 Kelas FormKeluarParkir

Pada FormKeluarParkir terdapat satu operasi yang berfungsi untuk mengirimkan seluruh data pemesanan yaitu *submitPemesanan().* Operasi ini mengirimkan data pemesanan pada kelas ControlPemesanan untuk kemudian dilakukan *insert* pada *database*.

3.7.4.39 Kelas FormListParkir

FormListParkir merupakan antarmuka dari proses pemesanan yang dilakukan oleh pengunjung. FormPemesanan merupakan tempat interaksi pengunjung dengan sistem dalam proses pengisian data pemesanan tiket.

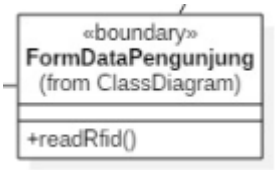


Gambar 3.90 Kelas FormListParkir

Pada FormListParkir terdapat satu operasi yang berfungsi untuk mengirimkan seluruh data pemesanan yaitu *submitPemesanan().* Operasi ini mengirimkan data pemesanan pada kelas ControlPemesanan untuk kemudian dilakukan *insert* pada *database*.

3.7.4.40 Kelas FormDataPengunjung

FormDataPengunjung merupakan antarmuka dari proses pemesanan yang dilakukan oleh pengunjung. FormPemesanan merupakan tempat interaksi pengunjung dengan sistem dalam proses pengisian data pemesanan tiket.

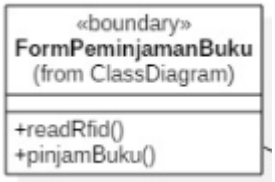


Gambar 3.91 Kelas FormDataPengunjung

Pada FormDataPengunjung terdapat satu operasi yang berfungsi untuk mengirimkan seluruh data pemesanan yaitu *submitPemesanan().* Operasi ini mengirimkan data pemesanan pada kelas ControlPemesanan untuk kemudian dilakukan *insert* pada *database*.

3.7.4.41 Kelas FormPeminjamanBuku

FormPeminjamanBuku merupakan antarmuka dari proses pemesanan yang dilakukan oleh pengunjung. FormPemesanan merupakan tempat interaksi pengunjung dengan sistem dalam proses pengisian data pemesanan tiket.

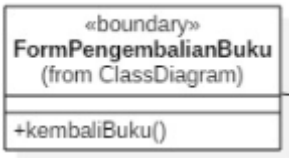


Gambar 3.92 Kelas FormPeminjamanBuku

Pada FormPeminjamanBuku terdapat satu operasi yang berfungsi untuk mengirimkan seluruh data pemesanan yaitu *submitPemesanan().* Operasi ini mengirimkan data pemesanan pada kelas ControlPemesanan untuk kemudian dilakukan *insert* pada *database*.

3.7.4.42 Kelas FormPengembalianBuku

FormPengembalianBuku merupakan antarmuka dari proses pengembalian buku yang dilakukan oleh pengunjung. FormPengembalianBuku merupakan tempat interaksi pengunjung dengan sistem dalam proses pengembalian.

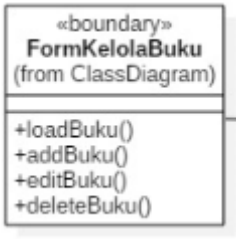


Gambar 3.93 Kelas FormPengembalianBuku

Pada FormPengembalianBuku terdapat satu operasi yang berfungsi untuk mengirimkan seluruh data pemesanan yaitu *kembaliBuku().* Operasi ini mengirimkan data pengembalian pada kelas Controltransaksi untuk kemudian dilakukan *insert* pada *database*.

3.7.4.43 Kelas FormKelolaBuku

FormKelolaBuku merupakan antarmuka dari proses kelola buku yang dilakukan oleh admin perpus.

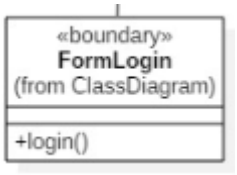


Gambar 3.94 Kelas FormKelolaBuku

Pada FormKelolaBuku terdapat satu empat yang berfungsi untuk mengirimkan seluruh data kelola data buku

3.7.4.44 Kelas FormLogin

FormLogin merupakan antarmuka dari sistem untuk proses login administrator. FormLogin merupakan tempat interaksi administrator dengan sistem dalam hal autentifiksasi login.

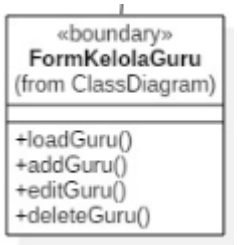


Gambar 3.95 Kelas FormLogin

Pada FromLogin terdapat satu operasi *btnLoginClick()* untuk autentifikasi pengguna. Satu operasi lainnya yaitu *btnLogoutClick()*untuk fungsionalitas logout

3.7.4.45 Kelas Form KelolaGuru

Form KelolaGuru merupakan antarmuka dari proses input data guru yang dilakukan oleh Admin. Form KelolaGuru merupakan tempat interaksi Admin dengan sistem dalam proses pengisian data Guru.



Gambar 3.96 Kelas FormKelolaGuru

Pada Form KelolaGuru terdapat empat operasi yang berfungsi untuk mengirimkan seluruh data Guru yaitu *AddGuru(),* melihat data guru *loadGuru(),* Edita data guru yaitu *EditGuru()* dan hapus data guru *deleteGuru().* Operasi ini mengirimkan data pemesanan pada kelas *GuruController* untuk kemudian dilakukan *insert* pada *database*.

3.7.4.46 Kelas FormDaftarSIM

FormDaftarSIM merupakan antarmuka dari proses pendaftaran Siswa yang memiliki SIM yang dilakukan oleh Admin Sistem.

Pada FormDaftarSIM terdapat satu operasi yang berfungsi untuk mengirimkan seluruh data kepemilikan SIM yaitu *submitDaftarSIM()*. Operasi ini mengirimkan data pemesanan pada kelas ControlSiswa untuk kemudian dilakukan *insert* pada *database*.

3.8 Perancangan Web Service

Pada yang akan dibangun, terdapat sebuah web service yang berfungsi sebagai pusat dari seluruh proses yang ada. Web service ini terhubung langsung dengan database dan akan menyuplai data ke seluruh subsistem lainnya. Web service ini mengimplementasikan arsitektur REST ditambah dengan JWT (JSON Web Token) yang berfungsi untuk melakukan otorisasi request melalui token yang dikirim oleh client melalui http header. Adapun contoh gambaran proses arsitektur dari web service tersebut adalah :

Masuk Parkir

Untuk masuk parkir tahapannya adalah tap rfid, kemudian input nomor parkir lalu post data ke backend.

Masuk Parkir

```
endpoint: http://82610c6c.ngrok.io/api/parkir/masuk
http method: POST
request body:
{
    "rfid": {uid dari rfid, contoh: "0015981952"},
    "nopol": {nomor polisi, contoh {"nopol": "D 1234 XX"}
}
json response:
{
```

```
  "title": "Create",
  "status": "Success",
  "message": "Create successful",
  "code": 201
}
```

Untuk keluar ada 2 tahapan, pertama tap rfid untuk mendapatkan detail biaya parkir dengan memanggil endpoint showDetail kemudian setelah mendapatkan data, input nomor polisi kemdian submit ke backend dengan mempost data ke endpoint keluar.

Show Detail

endpoint: http://82610c6c.ngrok.io/api/parkir/detail/{uid dari rfid}

contoh endpoint: http://localhost:8090/api/parkir/detail/0015981952

http method: GET

request body: TIDAK ADA

json response:

```
{
  "title": "Load",
  "status": "Success",
  "message": "Load successful",
  "code": 200,
  "data": {
    "id": 3,
    "rfid": "0015981952",
    "jam_masuk": "2017-06-11 15:28:46",
    "jam_keluar": null,
    "biaya": 2000,
    "nopol": "D 1234 XX"
  }
}
```

Keluar Parkir

```
endpoint:  http://localhost:8090/api/parkir/keluar/{uid  dari
rfid}
Contoh
endpoint: http://localhost:8090/api/parkir/keluar/0015981952
http method: PUT
request body:
{
    "rfid": {uid dari rfid, contoh: "0015981952"},
    "nopol": {nomor polisi, contoh {"nopol": "D 1234 XX"}
}
json response:
{
  "title": "Create",
  "status": "Success",
  "message": "Create successful",
  "code": 201
}
```

Dari contoh *web service* parkir, alur prosesnya dapat diutarakan sebagai berikut.

1. Aplikasi *client* (subsistem) harus mengakses *endpoint login* terlebih dahulu untuk mendapatkan token dengan contoh *request* seperti berikut.

```
Http method: POST /login
Request header : Content-Type=application/json
Request body:
{
“username”:”admin-perpus”,
“password”:”admin”
}
```

Jika *login* berhasil maka *web service* akan memberikan respon yang berisi token dengan kode respon 200 dalan bentuk JSON. Jika tidak, maka web service akan memberikan respon berisi pesan kesalahan dengan kode response 401 dalan bentuk JSON.

2. Setelah mendapatkan token, aplikasi *client* dapat mengakses *endpoint* lainnya sesuai dengan hak akses yang dimiliki. Contohnya untuk aplikasi perpus yang ingin menampilkan data buku, aplikasi perpustakaan dapat melakukan *request* ke *web service* dengan contoh sebagai berikut.

```
Http method: GET /perpus/buku
Request header : Authorization=Bearer <token>
```

Kemudian JWT akan melakukan pemeriksaan terhadap token tersebut, apabila valid maka JWT akan memberikan informasi *user* tersebut ke *action controller*. Apabila token tidak valid, maka JWT akan mengembalikan pesan error ke service *endpoint* dan *web service* pun akan mengembalikan respon berisi pesan kesalahan dalan bentuk JSON.

3. Informasi *user* yang dinyatakan valid, akan diteruskan ke *action controller*. *Action controller* sendiri merupakan bagian dari *web service* yang melakukan operasi atau proses inti dari *web service* yang berhubungan dengan proses bisnis. Masih dengan contoh get data buku, *application controller* akan memparsing parameter yang ada pada *request* kemudian melakukan *query* atau memanggil kelas ORM untuk kemudian mengambil data dari *database*.

4. Setalah data didapatkan, kemudian *application controller* akan melakukan proses berikutnya ataupun melakukan *mapping* data sesuai dengan kebutuhan dan juga proses bisnis. Lalu data tersebut dikembalikan ke service *endpoint* untuk dikirimkan ke *client* sebagai respon dalam bentuk JSON.

3.9 Perancangan Basis Data

Perancangan basis data berguna untuk memenuhi kebutuhan-kebutuhan konten informasi dan menyediakan struktur informasi yang alami dan mudah dipahami. Perancangan basis data terdiri dari skema relasi, diagram relasi dan struktur file.

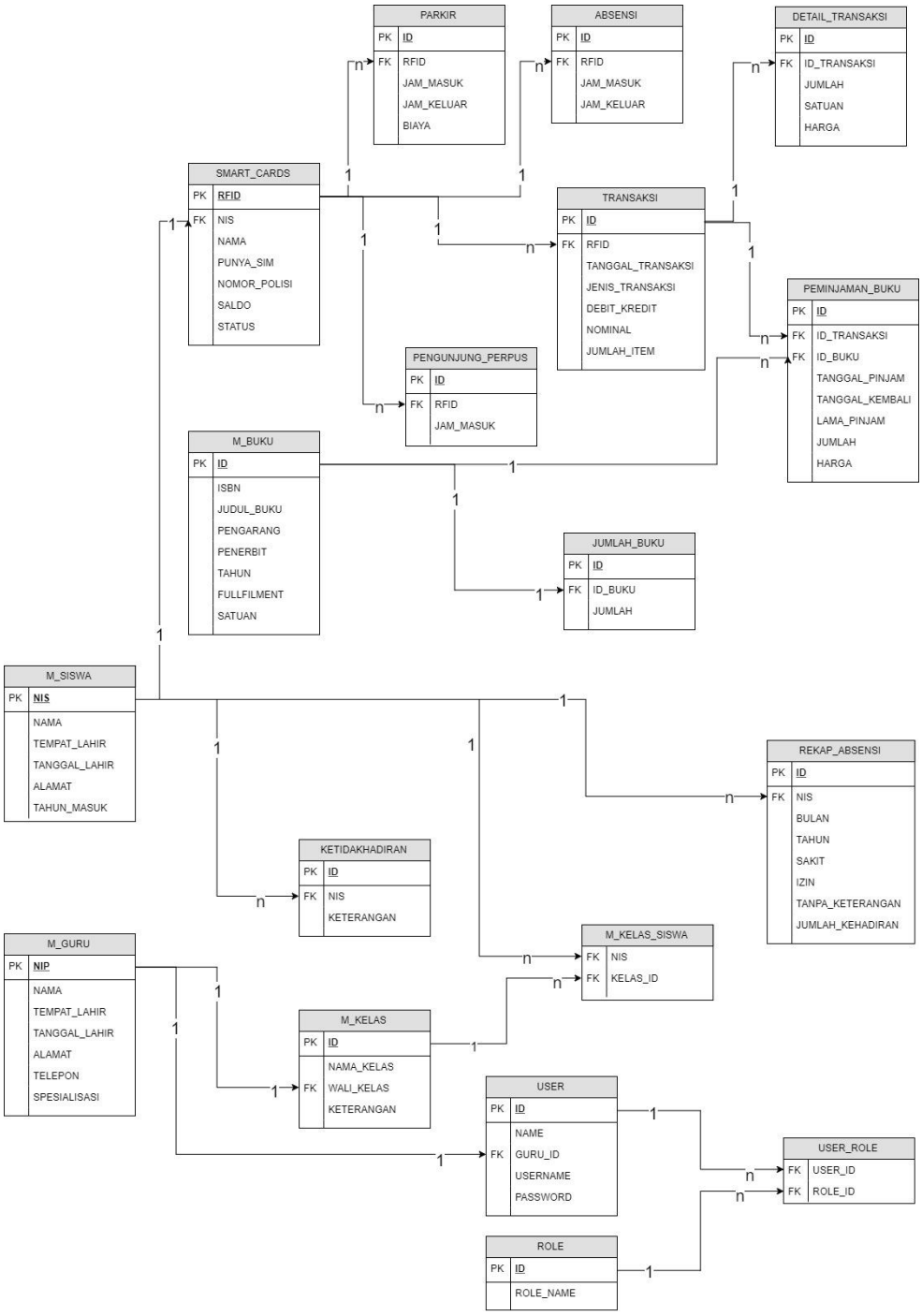
3.9.1 Skema Relasi

Skema relasi merupakan notasi untuk menggambar tabel, atribut tiap tabel dan hubungan atribut kunci antar tabel yang berelasi. Berikut adalah skema relasi dari basis data pada sistem yang dibangun.

1) Absensi
2) Buku
3) DetailPeminjaman
4) DetailTransaksi
5) Parkir
6) RekapitulasiAbsensi
7) SmartCard
8) Transaksi
9) Kelas
10) KelasSiswa
11) Guru
12) Siswa
13) UserRole
14) PengunjungPerpus
15) Ketidakhadiran
16) User
17) Role
18) JumlahBuku

3.9.2 Diagram Relasi

Diagram relasi merupakan diagram yang menjelaskan realasi antar tabel atau entitas. Gambar 3.56 adalah diagram relasi dari basis data pada sistem yang dibangun.



Gambar 3.97 Relasi Basis Data

3.9.3 Struktur Tabel

Struktur file menggambarkan file yang akan mewakili satu tabel. Dimana jumlah tabel maupun atribut pada struktur file harus sama dengan yang ada di skema relasi. Berikut adalah struktur file dari basis data pada sistem yang dikembangkan.

1) Absensi

Nama Field	Tipe	Panjang	Kunci	Keterangan
IDabsensi	int	-	Primary key	Auto increment, Unique
RFID	char		Foreign key	FK dari RFID pada tabel smartcard
jam_masuk	datetime	-	-	-
Jam_keluar	datetime	-	-	-

2) Buku

Nama Field	Tipe	Panjang	Kunci	Keterangan
IDbuku	int	-	Primary key	Auto increment, Unique
ISBN	varchar	13	-	-
Judul_buku	varchar	100	-	-
Pengarang	varchar	50	-	-
Penerbit	varchar	50	-	-
Tahun	int	4	-	-
Fullfilment	int	5	-	-
satuan	varchar	5	-	-

3) DetailPeminjaman

Nama Field	Tipe	Panjang	Kunci	Keterangan
IDDetailPeminjaman	int	-	Primary key	Auto increment

IDTransaksi	int	-	Foreign key	FK dari IDTransaksi pada tabel DetailTransaksi
IDBuku	int	-	Foreign key	FK dari IDBuku pada tabel Buku
Tanggal_pinjam	datetime	-	-	-
Tanggal_kembali	datetime	-	-	-
Lama_pinjam	int	5	-	-
Jumlah	Int	3	-	-
Harga	Int	10	-	-

4) DetailTransaksi

Nama Field	Tipe	Panjang	Kunci	Keterangan
IDDetailTransaksi	int	-	Primary key	Auto increment, Unique
IDTransaksi	int		Foreign key	FK dari IDtransaksi pada tabel transaksi
Jumlah	Int	-	-	-
Satuan	varchar	-	-	-
Harga	int	10	-	-

5) Parkir

Nama Field	Tipe	Panjang	Kunci	Keterangan
IDParkir	int	-	Primary key	Auto increment, Unique
RFID	char		Foreign key	FK dari RFID pada tabel smartcard
jam_masuk	datetime	-	-	-
Jam_keluar	datetime	-	-	-
Biaya	Int	-	-	-

6) RekapAbsensi

Nama Field	Tipe	Panjang	Kunci	Keterangan
IDRekap	int	-	Primary key	Auto increment, Unique
nis	char	10	Foreign key	FK dari nis pada tabel siswa
bulan	datetime	2	-	-
tahun	int	4	-	-
sakit	int	3	-	-
izin	int	3	-	-
Tanpa_keterangan	int	3	-	-
Jumlah_kehadiran	int	3	-	-

7) SmartCard

Nama Field	Tipe	Panjang	Kunci	Keterangan
RFID	char	10	Primary key	Auto increment, Unique
nis	char	10	Foreign key	FK dari nis pada tabel siswa
nama	varchar	100	-	-
Punya_sim	boolean	-	-	-
saldo	int	10	-	-
Status	Varchar	15	-	-
Nomor_polisi1	varchar	8	-	-
Nomor_polisi2	varchar	8	-	-
Nomor_polisi3	varchar	8	-	-
Nomor_polisi4	varchar	8	-	-
Nomor_polisi5	varchar	8	-	-

8) Transaksi

Nama Field	Tipe	Panjang	Kunci	Keterangan

IDTransaksi	int	-	Primary key	Auto increment, Unique
RFID	char		Foreign key	FK dari RFID pada tabel smartcard
Tanggal_transaksi	datetime	-	-	-
Jenis_transaksi	varchar	20	-	-
Debit_kredit	varchar	20	-	-
Jumlah	int	10	-	-
Nominal	varchar	50	-	-

9) Kelas

Nama Field	Tipe	Panjang	Kunci	Keterangan
IDKelas	int	-	Primary key	Auto increment, Unique
Nama_kelas	varchar	20	-	-
Wali_kelas	datetime	-	Foreign key	FK dari guru pada tabel guru
Keterangan	varchar	-	-	-

10) KelasSiswa

Nama Field	Tipe	Panjang	Kunci	Keterangan
Siswa_ID	int	-	Foreign Key	FK dari nis pada tabel kelas
Kelas_ID	int	-	Foreign key	FK dari IDKelas pada tabel siswa

11) Guru

Nama Field	Tipe	Panjang	Kunci	Keterangan
Nip	char	10	Primary key	Unique
Nama	varchar	100	Foreign key	FK dari RFID pada tabel smart_cards

Tempat_lahir	varchar	20	-	-
Tanggal_lahir	datetime	-	-	-
Alamat	Varchar	100	-	-
Telepon	Int	20	-	-
Spesialisasi	Varchar	100	-	-

12) Siswa

Nama Field	Tipe	Panjang	Kunci	Keterangan
Nis	int	10	Primary key	Auto increment
Nama	Varchar	100	-	-
Tempat_lahir	Varchar	20	-	-
Tanggal_lahir	datetime	-	-	-
Alamat	Varchar	100	-	-
Tahun_masuk	Int	4	-	-

13) UserRole

Nama Field	Tipe	Panjang	Kunci	Keterangan
User_ID	int	-	Foreign key	FK dari IDuser pada tabel user
Role_ID	int		Foreign key	FK dari IDrole pada tabel role

14) PengunjungPerpus

Nama Field	Tipe	Panjang	Kunci	Keterangan
IDpengunjungperpus	int	-	Primary key	Auto increment, Unique
RFID	string		Foreign key	FK dari RFID pada tabel smartcard
jam_masuk	datetime	-	-	-

15) Ketidakhadiran

Nama Field	Tipe	Panjang	Kunci	Keterangan
IDabsensi	int	-	Primary key	Auto increment, Unique
RFID	string		Foreign key	FK dari RFID pada tabel smart_cards
jam_masuk	datetime	-	-	-
Jam_keluar	datetime	-	-	-

16) User

Nama Field	Tipe	Panjang	Kunci	Keterangan
IDuser	int	-	Primary key	Auto increment, Unique
Nama	varchar	100	-	
Guru_ID	int	-	Foreign key	FK dari guru pada tabel guru
Username	varchar	15	-	-
password	Varchar	30	-	-

17) Role

Nama Field	Tipe	Panjang	Kunci	Keterangan
IDRole	int	-	Primary key	Auto increment, Unique
Role_name	varchar	-	-	-

18) JumlahBuku

Nama Field	Tipe	Panjang	Kunci	Keterangan
IDJumlahBuku	int	-	Primary key	Auto increment, Unique
Buku_ID	int		Foreign key	FK dari IDBuku pada tabel buku
jumlah	int	4	-	-

3.10 Perancangan Sistem

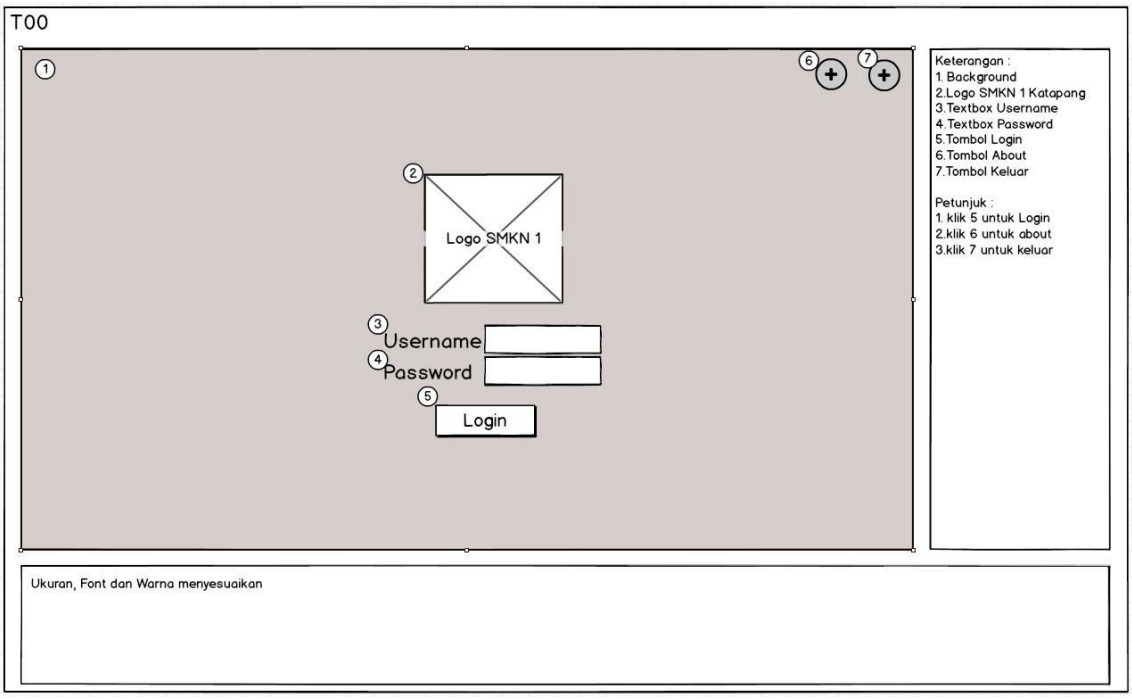
Pada tahap ini perancangan akan didefinisikan secara detail yang berkaitan dengan kegiatan implementasi seperti perancangan menu, perancangan anatarmuka, dan jaringan semantik.

3.10.1 Perancangan Antar Muka

Perancangan antarmuka merupakan perancangan dari antarmuka sistem dimana antamuka merupakan tempat sistem berinteraksi dengan pengguna. Dalam perancangan antarmuka sistem ini, terdapat dua jenis antarmuka, yaitu antarmuka form atau halaman dan antarmuka pesan.

3.10.1.1 Antarmuka Halaman Login Pengguna

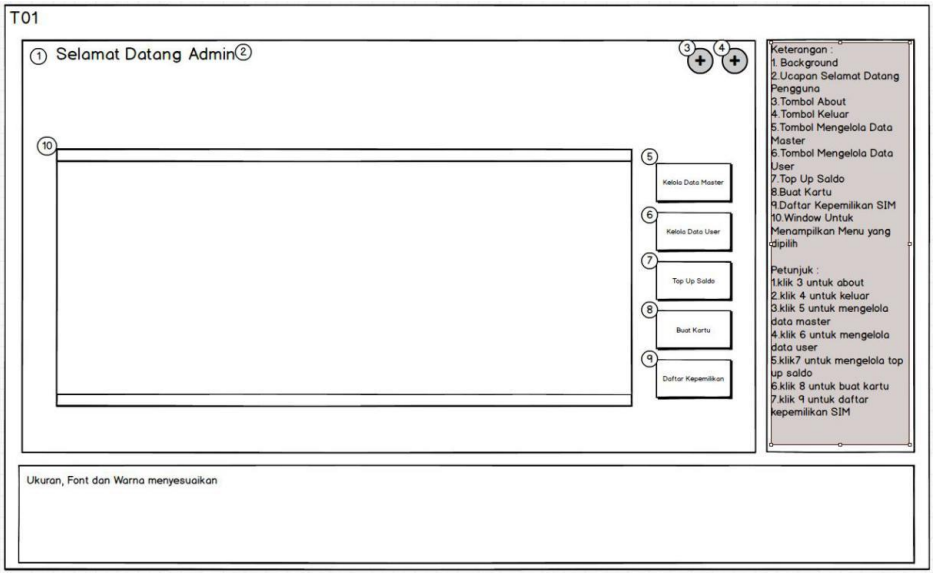
Berikut rancangan awal untuk halaman antarmuka login, dapat dilihat pada gambar 3.107.



Gambar 3.98 Antarmuka Login

3.10.1.2 Halaman Antarmuka Admin

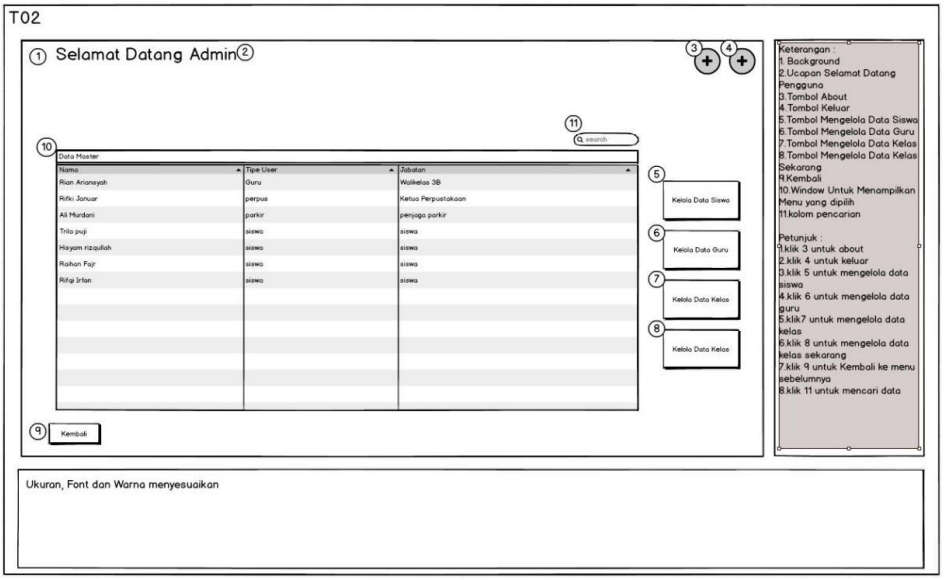
Berikut rancangan awal untuk halaman antarmuka admin, dapat dilihat pada gambar 3.108.



Gambar 3.99 Antarmuka Admin

3.10.1.3 Halaman Antarmuka Kelola Data Master

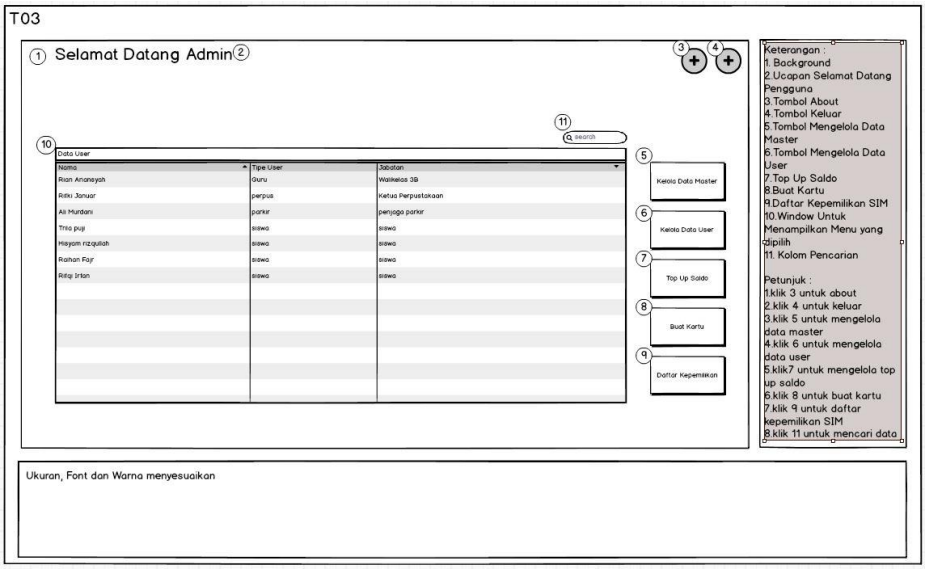
Berikut rancangan awal untuk halaman antarmuka kelola data master, dapat dilihat pada gambar 3.109.



Gambar 3.100 Antarmuka Kelola Data Master

3.10.1.4 Halaman Antarmuka Data User

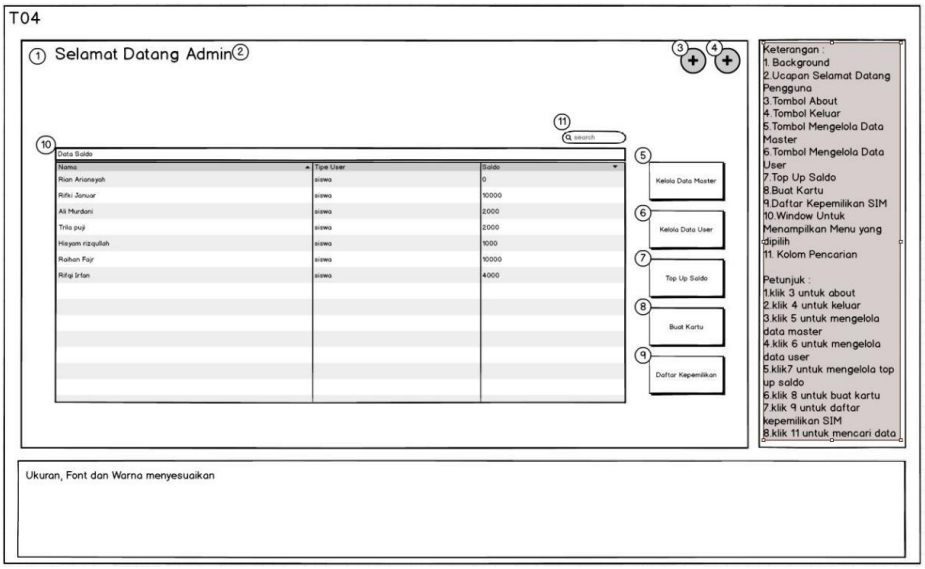
Berikut rancangan awal untuk halaman antarmuka data user, dapat dilihat pada gambar 3.110



Gambar 3.101 Antarmuka Data User

3.10.1.5 Halaman Antarmuka Data Saldo

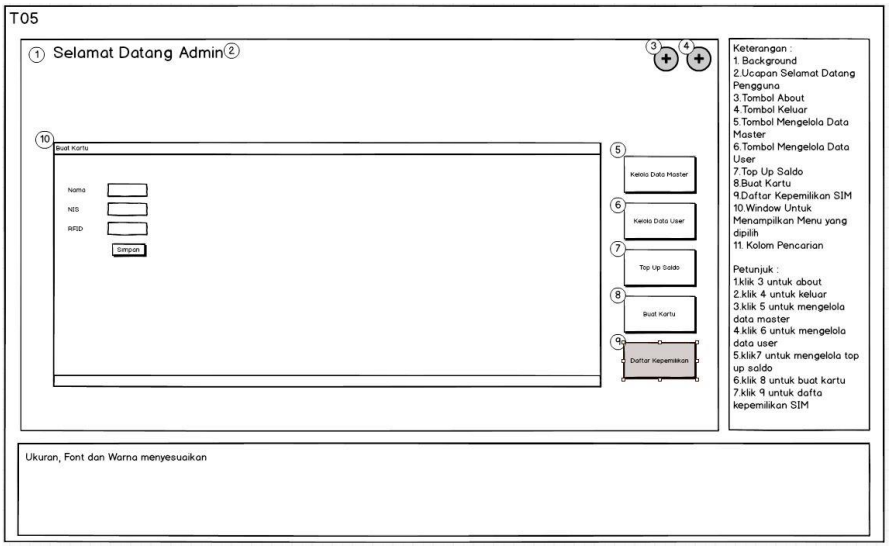
Berikut rancangan awal untuk halaman antarmuka data saldo, dapat dilihat pada gambar 3.111.



Gambar 3.102 Antarmuka Data Saldo

3.10.1.6 Halaman Antarmuka Data Kartu

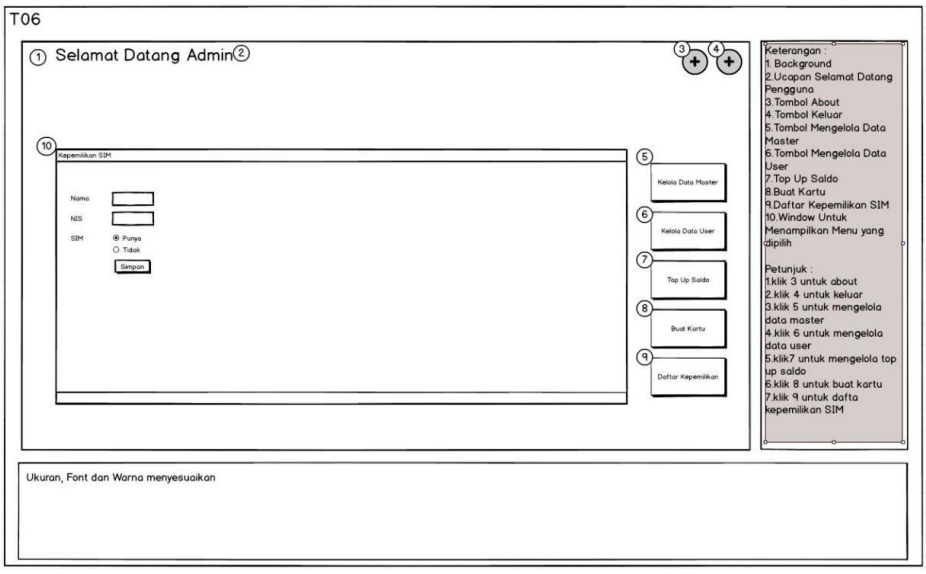
Berikut rancangan awal untuk halaman antarmuka data kartu, dapat dilihat pada gambar 3.112.



Gambar 3.103 Antarmuka Data Kartu

3.10.1.7 Halaman Antarmuka Data Kepemilikan SIM

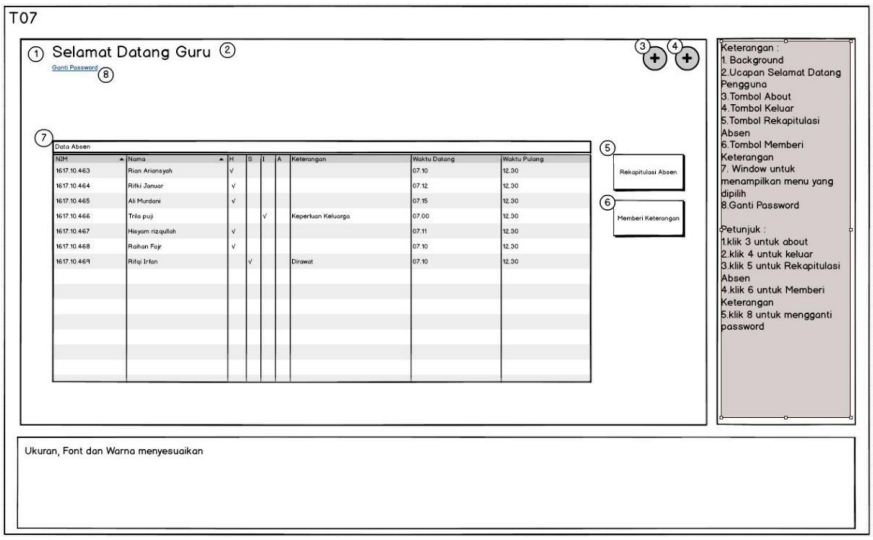
Berikut rancangan awal untuk halaman antarmuka kepemilikan SIM, dapat dilihat pada gambar 3.113.



Gambar 3.104 Antarmuka Data Kepemilikan SIM

3.10.1.8 Halaman Antarmuka Guru

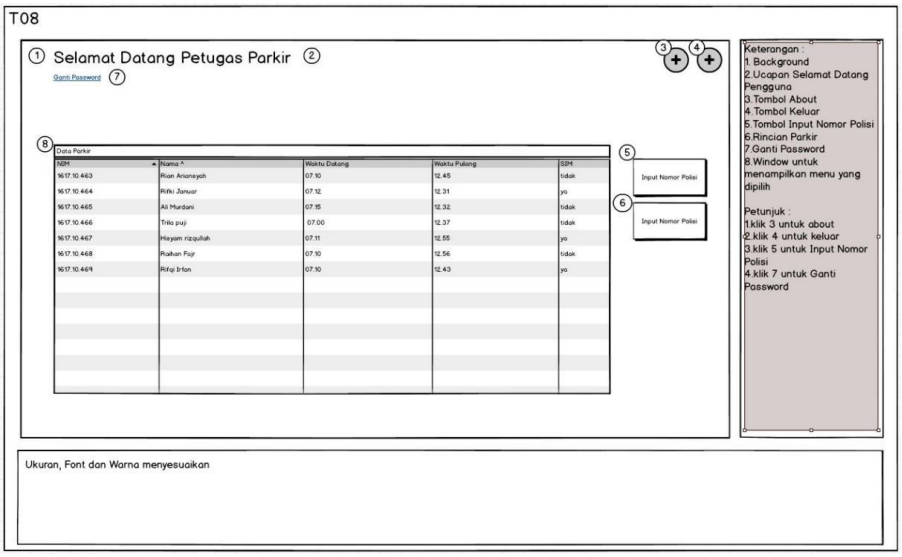
Berikut rancangan awal untuk halaman antarmuka guru, dapat dilihat pada gambar 3.114.



Gambar 3.105 Antarmuka Guru

3.10.1.9 Halaman Antarmuka Petugas Parkir

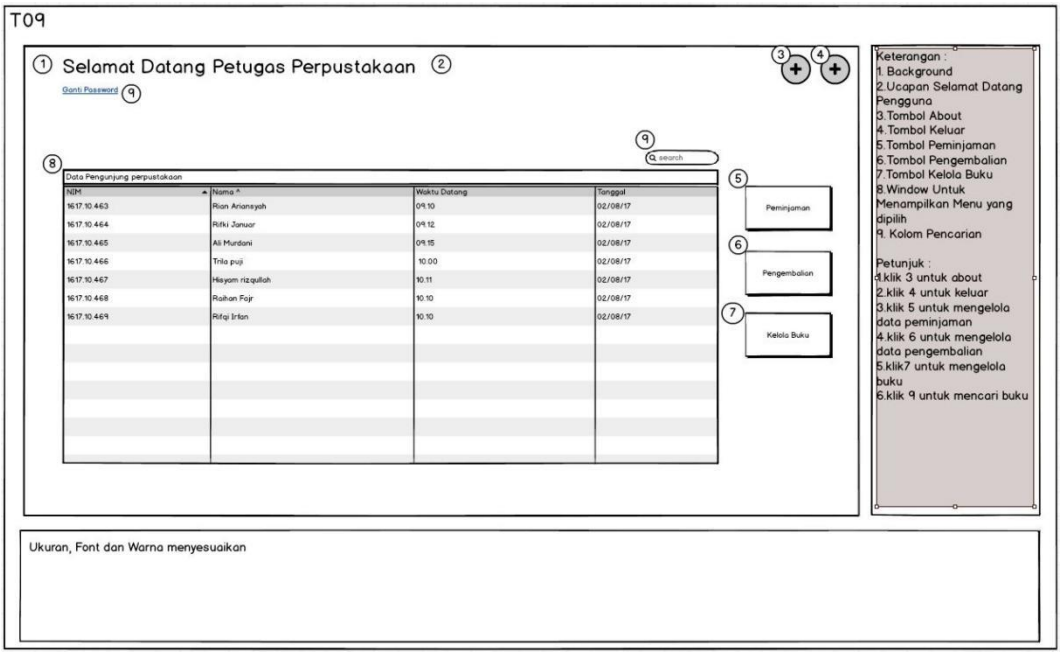
Berikut rancangan awal untuk halaman antarmuka petugas parkir, dapat dilihat pada gambar 3.115.



Gambar 3.106 Antarmuka Petugas Parkir

3.10.1.10 Halaman Antarmuka Petugas Perpustakaan

Berikut rancangan awal untuk halaman antarmuka petugas perpustakaan, dapat dilihat pada gambar 3.116.



Gambar 3.107 Petugas Perpustakaan