

BAB 4

HASIL DAN PEMBAHASAN

Pada bab ini disajikan hasil dari setiap tahapan dalam metodologi penelitian yang telah dilaksanakan. Mengacu pada langkah-langkah metodologis yang diterapkan, pembahasan mencakup hasil dari tahapan Identifikasi Masalah, Study Literatur, Requirement Analysis, System Design, Implementation, Testing, Maintenance, dan Development.

4.1 Identifikasi Masalah

Pada tahap identifikasi masalah, dilakukan analisis terhadap isu-isu yang melatarbelakangi kebutuhan Pencucian A&R akan suatu sistem informasi. Analisis permasalahan ini didasarkan pada data yang dikumpulkan melalui observasi terhadap alur kerja serta pengumpulan data transaksi guna mengidentifikasi jenis data yang diperlukan. Selain itu, metode wawancara digunakan untuk memperoleh informasi secara lebih mendalam dari para pemangku kepentingan (stakeholder).

Wawancara dilaksanakan secara daring dengan Ibu Jaliah selaku pemilik usaha Pencucian A&R. Dalam wawancara tersebut dibahas kondisi operasional terkini, di mana teridentifikasi beberapa permasalahan, antara lain proses transaksi yang masih dilakukan secara manual, kendala dalam pencatatan dan penyimpanan data transaksi, serta ketidakteraturan dalam pencatatan kehadiran karyawan. Untuk memperoleh pemahaman yang lebih mendalam serta sebagai dasar dalam merancang solusi terhadap permasalahan tersebut, maka diperlukan kajian literatur yang membahas teori-teori, konsep, dan penelitian terdahulu yang relevan.

4.2 Studi Literatur

Setelah melakukan identifikasi masalah selanjutnya dilakukan tahap studi literatur. Pada tahap ini akan dilakukan telaah terhadap berbagai sumber referensi yang relevan dengan permasalahan yang diangkat, serta berpotensi mendukung

pencapaian tujuan penelitian. Rangkuman dari sejumlah penelitian terdahulu yang memiliki kesesuaian konteks maupun permasalahan dengan penelitian ini disajikan pada Tabel 2. .

Perancangan dan pembangunan aplikasi ini mengacu pada metode *System Development Life Cycle* (SDLC), dengan menggunakan salah satu model pengembangannya, yaitu model *Waterfall*. Dalam penerapan model *Waterfall*, dihasilkan sejumlah dokumen penting seperti Spesifikasi Kebutuhan Perangkat Lunak (SKPL), Desain Perangkat Lunak (DPPL), dan Dokumen Uji Perangkat Lunak (DUPL). Masing-masing dokumen tersebut menggunakan sistem penomoran tertentu serta mencantumkan definisi dan terminologi yang sesuai dengan konteks pengembangan sistem. Setelah dilakukan studi literatur maka akan dilanjutkan pada tahapan Requirement Analysis.

4.3 Requirement Analysis

Setelah dilakukan kajian terhadap berbagai referensi pada tahap studi literatur, diperoleh landasan teori dan konsep yang menjadi dasar dalam proses perancangan sistem. Berdasarkan pemahaman tersebut, langkah berikutnya adalah melakukan analisis kebutuhan (requirement analysis) untuk mengidentifikasi fitur-fitur serta spesifikasi yang diperlukan baik oleh pengguna maupun oleh sistem secara menyeluruh. Untuk mengidentifikasi fitur apa saja yang dibutuhkan maka, dilakukan wawancara kepada *stakeholder*. Adapun Notulensi wawancara yang dilakukan seperti pada tabel 4.1 :

Tabel 4.1 Notulensi Wawancara

No	Pertanyaan	Jawaban
1	Apakah permasalahan terbesar yang terjadi dalam Pencucian A&R ?	Akhir-akhir ini pemasukan pencucian tidak sinkron antara kendaraan yang dicuci dengan buku catatan transaksi. Dan juga karyawan

		sering curang dalam kehadiran, seperti dibasensi ada tapi kenyataan di tempat kerja orangnya tidak ada. Untuk pencatatan transaksi dan data pelanggan kami masih dilakukan secara manual, sehingga menyulitkan kami dalam memantau performa harian, membuat laporan keuangan, atau melakukan promosi yang terarah.
2	Berdasarkan masalah yang terjadi apakah berpengaruh besar terhadap performa pencucian dalam melayani pelanggan?	Dari apa yang terjadi sekarang cukup berpengaruh ya. Sebagai contoh untuk karyawan yang manipulasi kehadiran sangat berpengaruh terhadap durasi waktu yang dibutuhkan untuk mencuci 1 kendaraan. Normalnya 5 orang dapat menyelesaikan 1 kendaraan dalam waktu 15-20 menit sudah termasuk proses pencucian hingga pengeringan. Namun jika yang bekerja hanya 3-4 orang waktu yang dibutuhkan bisa hingga 30-40 menit. Dengan selisih waktu tersebut bisa membuat penghasilan dari pencucian kurang maksimal.
3	Lalu bagaimana dengan pencatatan transaksi yang masih dilakukan dengan manual menggunakan buku catatan?	Kami sudah 2 kali mengganti buku yang digunakan untuk mencatat. Bukan karena penuh tapi karena terkena cipratan air dari karyawan yang juga melakukan proses pencucian kendaraan. buku yang terkena cipratan air itu terkadang menghilangkan tulisan yang ada di buku. Sehingga itu sedikit mengganggu ketika perhitungan pemasukan di akhir bulan terutama untuk penggajian.
4	Ada berapa total karyawan di Pencucian A&R? dan bagaimana sistem kerja dari karyawan	Total karyawan ada 10 orang. 6 orang mengerjakan proses pencucian, Tiga orang di

	tersebut?	pencucian mengerjakan 1 kendaraan, sehingga kendaraan yang dapat dikerjakan sekaligus bisa hingga 2 kendaraan. 4 orang lainnya mengerjakan proses pengeringan, dengan sistem yang sama per dua orang mengerjakan 1 kendaraan.
5	Sebelumnya apakah pernah terjadi manipulasi data transaksi yang dilakukan karyawan? mengingat sebelumnya dikatakan data transaksi yang ada dibuku pernah hilang karena terkena air	Pernah dan ketahuan walaupun yang ketahuan baru terjadi 2 kali. mungkin bisa jadi sebelumnya pernah dilakukan namun belum ketahuan.
6	Apakah pencucian A&R memerlukan sebuah aplikasi yang dapat mengelola data transaksi dan karyawan?	Untuk sekarang perlu mengingat data transaksi yang sering hilang sehingga rawan dimanipulasi oleh karyawan sendiri

Berdasarkan tabel 4.1 didapatkan mengenai kondisi yang terjadi di pencucian A&R terkait kondisi aktivitas pencatatan transaksi yang masih manual dan sering terjadinya manipulasi data yang dilakukan oleh karyawan. Oleh karena itu diperlukan sebuah sistem yang dapat menyelesaikan permasalahan yang ada di pencucian A&R.

Dalam proses pengembangan sistem informasi pencucian A&R, disusun dokumen yang berfungsi sebagai acuan dalam pengembangan perangkat lunak, yang merinci berbagai kebutuhan sistem secara lengkap. Melalui pendekatan Waterfall, salah satu dokumen yang dihasilkan adalah Software Requirements Specification (SRS) atau Spesifikasi Kebutuhan Perangkat Lunak, yang memuat deskripsi terperinci mengenai fungsionalitas dan spesifikasi sistem yang akan dikembangkan. Dokumen SRS ini dapat ditemukan pada Lampiran B dan merupakan bagian dari dokumen pendukung dalam perancangan dan pembangunan Sistem Informasi Pencucian A&R.

4.3.1 Kebutuhan Fungsional Perangkat Lunak

Fitur-fitur yang tersedia pada perangkat lunak, yang juga dikenal sebagai kebutuhan fungsional sistem, merupakan komponen yang mencerminkan seluruh proses atau layanan yang akan disediakan sistem kepada pengguna. Adapun daftar kebutuhan fungsional yang terdapat pada Sistem Informasi Pencucian A&R (SIPAR) adalah sebagai berikut:

A. Pengguna : Admin

Tabel 4.2 Daftar kebutuhan fungsional untuk pengguna admin

No	Kode Fungsi	Nama Fungsi
1.	SRS_F-SIPAR-002	<i>Login</i>
2.	SRS_F-SIPAR-003	<i>Logout</i>
3.	SRS_F-SIPAR-004	Menambah Data Karyawan
4.	SRS_F-SIPAR-005	Mengubah data keuangan
5.	SRS_F-SIPAR-006	Menghapus data keuangan
6.	SRS_F-SIPAR-007	Menambah data keuangan
7.	SRS_F-SIPAR-008	Mengubah data karyawan
8.	SRS_F-SIPAR-009	Menghapus data karyawan
9.	SRS_F-SIPAR-010	Melihat data rekapan keuangan

Adapun penjelasan dan rincian fitur diatas yaitu:

1. Administrator dapat mengakses aplikasi SIPAR melalui fungsi Login.
2. Administrator dapat menutup aplikasi SIPAR melalui fungsi Logout.
3. Administrator dapat menambahkan karyawan beserta jadwal karyawan tersebut.
4. Administrator dapat mengubah data keuangan dalam aplikasi SIPAR melalui fungsi ini.

5. Administrator dapat menghapus data keuangan dalam aplikasi SIPAR melalui fungsi ini.
6. Administrator dapat menambah data keuangan dalam aplikasi SIPAR melalui fungsi ini
7. Administrator dapat mengubah data karyawan dalam aplikasi SIPAR melalui fungsi ini.
8. Administrator dapat menghapus data karyawan dalam aplikasi SIPAR melalui fungsi ini.
9. Administrator dapat melihat data rekapan keuangan dalam aplikasi SIPAR melalui

B. Pengguna : Customer

Tabel 4.3 Daftar kebutuhan fungsional untuk pengguna customer

No	Kode Fungsi	Nama Fungsi
1.	SRS_F-SIPAR-001	Register
2.	SRS_F-SIPAR-002	Login
3.	SRS_F-SIPAR-003	Logout
4.	SRS_F-SIPAR-011	Order

Adapun penjelasan dan rincian fitur diatas yaitu:

1. Customer dapat mengakses aplikasi SIPAR melalui fungsi Register.
2. Customer bisa mengakses aplikasi SIPAR melalui fungsi Login.
3. Customer bisa menutup aplikasi SIPAR melalui fungsi Logout.
4. Customer dapat memesan jasa cuci mobil di aplikasi SIPAR melalui fungsi Order.

C. Pengguna : Karyawan

Tabel 4.4 Daftar kebutuhan fungsional untuk pengguna karyawan

No	Kode Fungsi	Nama Fungsi
1.	SRS_F-SIPAR-002	Login
2.	SRS_F-SIPAR-003	Logout
3.	SRS_F-SIPAR-012	Melihat jadwal karyawan
4.	SRS_F-SIPAR-013	Memasukkan data transaksi

Adapun penjelasan dan rincian fitur diatas yaitu:

1. Karyawan dapat mengakses aplikasi SIPAR melalui fungsi Login.
2. Karyawan dapat menutup aplikasi SIPAR melalui fungsi Logout.
3. Karyawan dapat melihat jadwal karyawan melalui fungsi ini.
4. Karyawan dapat memasukkan data hasil transaksi jasa pencucian kendaraan ke dalam aplikasi SIPAR melalui fungsi ini.

4.3.2 Kebutuhan Non-Fungsional Perangkat Lunak

Kebutuhan non-fungsional merupakan spesifikasi yang berfokus pada karakteristik perilaku dan kualitas sistem, seperti kinerja, keamanan, keandalan, dan kemudahan penggunaan. Adapun daftar kebutuhan non-fungsional dari perangkat lunak Sistem Informasi Pencucian A&R (SIPAR) yang disesuaikan berdasarkan kategori pengguna adalah sebagai berikut:

Tabel 4.5 Kebutuhan Fungsional Perangkat Lunak

No	Kode	Parameter	Deskripsi Kebutuhan
1.	SRS_NF-SIPAR-001	Kapasitas	Aplikasi SIPAR dapat menyimpan data – data usaha meliputi rekapan data keuangan, data karyawan dan data customer.

2.	SRS_NF-SIP AR-002	Keamanan	Aplikasi SIPAR dilengkapi dengan fitur Login, dimana pengguna tidak dapat menggunakan aplikasi jika tidak memiliki akun.
3.	SRS_NF-SIP AR-003	Performa	Aplikasi SIPAR dapat memberikan pelayanan atau informasi secara cepat dan akurat
4.	SRS_NF-SIP AR-004	Kehandalan	Aplikasi SIPAR dapat diakses secara bersamaan asalkan bandwidth dari hostingnya tidak penuh.
5.	SRS_NF-SIP AR-005	Kemudahan	Aplikasi SIPAR dapat diakses dengan gampang dikarenakan User Experience yang mudah dipahami.
6.	SRS_NF-SIP AR-006	Kompabilitas	Aplikasi SIPAR dapat diakses menggunakan beberapa <i>browser</i> seperti Google Chrome, Mozilla Firefox, Safari, Opera dan lain sebagainya.

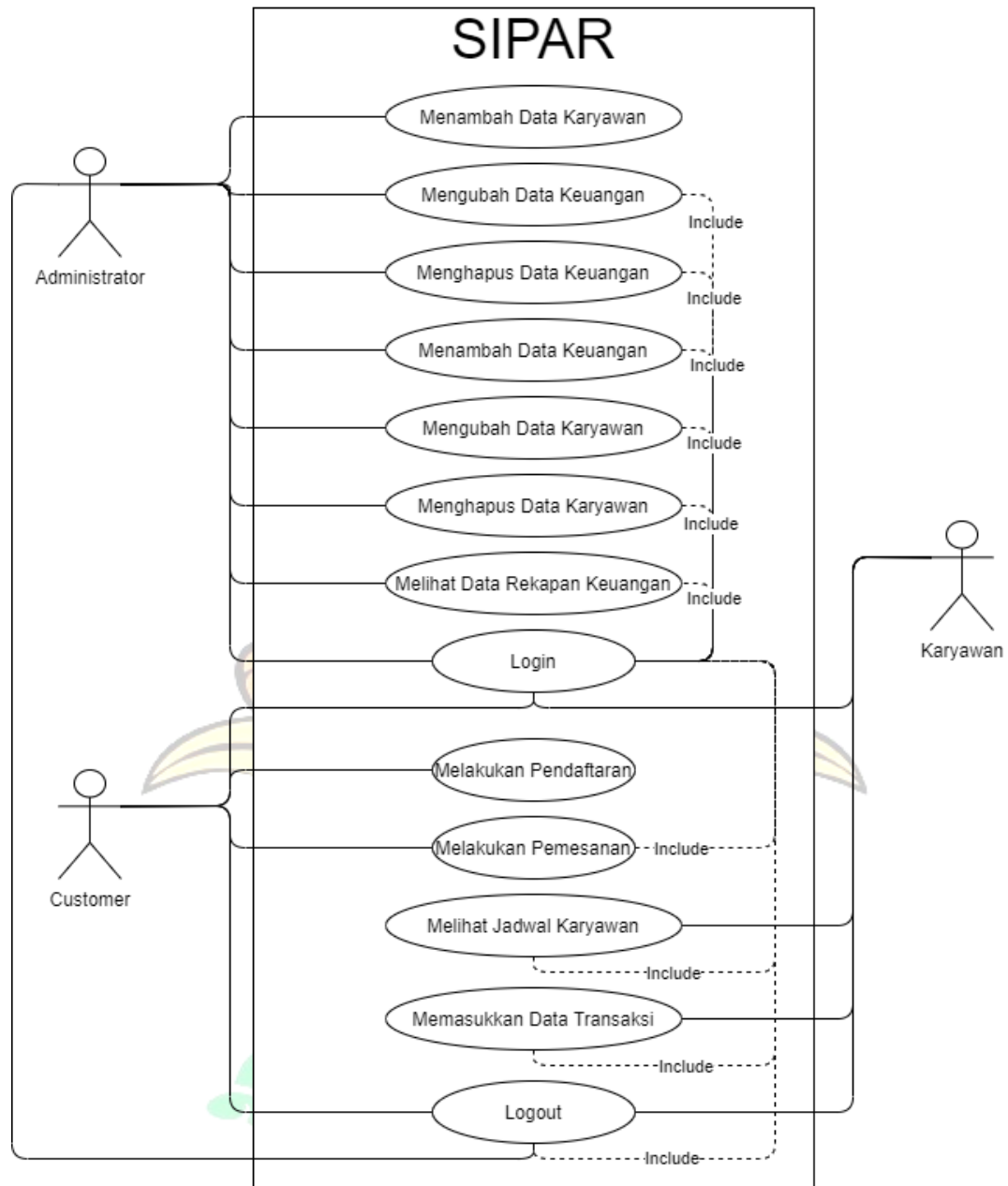
Adapun penjelasan dan rincian fitur diatas yaitu:

1. Aplikasi SIPAR dapat menyimpan data – data usaha meliputi rekapan data keuangan, data karyawan dan data customer.
2. Aplikasi SIPAR dilengkapi dengan fitur Login, dimana pengguna tidak dapat menggunakan aplikasi jika tidak memiliki akun.
3. Aplikasi SIPAR dapat memberikan pelayanan atau informasi secara cepat dan akurat
4. Aplikasi SIPAR dapat diakses secara bersamaan asalkan bandwidth dari hostingnya tidak penuh.
5. Aplikasi SIPAR dapat diakses dengan gampang dikarenakan User Experience yang mudah dipahami.
6. Aplikasi SIPAR dapat diakses menggunakan beberapa browser seperti Google Chrome, Mozilla Firefox, Safari, Opera dan lain sebagainya.

4.3.3 Use Case Diagram

Use Case class digunakan untuk memodelkan dan menyatakan unit fungsi/ layanan yang disediakan oleh sistem. Use

case adalah sesuatu yang menyediakan hasil yang dapat diukur ke pemakai atau sistem eksternal.



Gambar 4.1 Use Case Diagram

Pada gambar 4.1 merupakan sebuah use case diagram dari SIPAR. Dalam use case diagram tersebut terdapat 3 aktor yaitu Administrator, Karyawan, dan Costumer dengan total 13 use case.

4.3.5 Scenario Use Case

Use Case scenario adalah contoh fitur dari sebuah use case, untuk sistem aplikasi SIPAR. Dibawah ini merupakan beberapa scenario use case utama yang ada pada SIPAR.

Tabel 4.6 Skenario use case Menambah data karyawan

Nama Usecase	Menambah data karyawan
Kode	SRS_F-SIPAR-004
Actor	Administrator
Deskripsi	Administrator dapat menambahkan data dan jadwal dari karyawan
Pre-kondisi	Administrator telah login ke sistem
Main Flow	1. Administrator mengakses website SIPAR 2. Administrator mengklik menu data karyawan 3. Administrator menambahkan data karyawan dan menambahkan jadwal dari karyawan tersebut ke dalam sistem
Alternative flow	Jika Administrator menekan “Cancel” maka proses akan dibatalkan dan tampilan akan kembali ke menu data karyawan
Post-kondisi	Data dari karyawan telah tersimpan di dalam sistem

Tabel 4.7 Skenario use case Mengubah data keuangan

Nama Usecase	Mengubah data keuangan
Kode	SRS_F-SIPAR-005
Actor	Administrator

Deskripsi	Administrator dapat melakukan perubahan pada data keuangan dalam sistem aplikasi SIPAR
Pre-kondisi	Administrator telah login ke sistem sebagai administrator dan data keuangan telah sebelumnya tersimpan ke dalam sistem
Main Flow	1. Administrator mengakses website SIPAR 2. Administrator mengklik menu data keuangan 3. Administrator memilih opsi mengubah data keuangan 4. Administrator mengubah data keuangan sesuai dengan kebutuhannya
Alternative flow	Jika Administrator menekan “Cancel” maka proses akan dibatalkan dan tampilan akan kembali ke menu keuangan
Post-kondisi	Perubahan data keuangan telah tersimpan dalam sistem

Tabel 4.8 Skenario use case Menghapus data keuangan

Nama Usecase	Menghapus data keuangan
Kode	SRS_F-SIPAR-006
Actor	Administrator
Deskripsi	Administrator dapat melakukan penghapusan pada data keuangan dalam sistem aplikasi SIPAR
Pre-kondisi	Administrator telah login ke sistem atau data keuangan telah sebelumnya tersimpan ke dalam sistem
Main Flow	1. Administrator mengakses website SIPAR

	- Administrator mengklik menu data keuangan - Administrator memilih opsi menghapus data keuangan - Administrator menghapus data keuangan sesuai dengan kebutuhannya
Alternative flow	Jika Administrator menekan “Cancel” maka proses akan dibatalkan dan tampilan akan kembali ke menu keuangan
Post-kondisi	Perubahan berupa penghapusan data keuangan telah tersimpan

Tabel 4.9 Skenario use case Menambah data keuangan

Nama Usecase	Menambah data keuangan
Kode	SRS_F-SIPAR-007
Actor	Administrator
Deskripsi	Administrator dapat melakukan penambahan pada data keuangan dalam sistem aplikasi SIPAR
Pre-kondisi	Administrator telah login ke sistem, belum ada data keuangan yang tersimpan dalam sistem dan data keuangan telah sebelumnya tersimpan ke dalam sistem
Main Flow	- Administrator mengakses website SIPAR - Administrator mengklik menu data keuangan - Administrator memilih opsi mengubah data keuangan - Administrator mengubah data keuangan sesuai dengan kebutuhannya

Alternative flow	Jika Administrator menekan “Cancel” maka proses akan dibatalkan dan tampilan akan kembali ke menu keuangan
Post-kondisi	Perubahan berupa penambahan data keuangan telah tersimpan

Tabel 4.10 Skenario use case Mengubah data karyawan

Nama Usecase	Mengubah data karyawan
Kode	SRS_F-SIPAR-008
Actor	Administrator
Deskripsi	Administrator dapat melakukan perubahan pada data karyawan dalam sistem aplikasi SIPAR
Pre-kondisi	Administrator telah login ke sistem serta data karyawan telah sebelumnya tersimpan ke dalam sistem
Main Flow	1. Administrator mengakses website SIPAR 2. Administrator mengklik menu data karyawan 3. Administrator memilih opsi mengubah data karyawan 4. Administrator mengubah data karyawan sesuai dengan kebutuhannya
Alternative flow	Jika Administrator menekan “Cancel” maka proses akan dibatalkan dan tampilan akan kembali ke menu karyawan
Post-kondisi	Perubahan pada data karyawan telah tersimpan dalam sistem

Tabel 4.11 Skenario use case Menghapus data karyawan

Nama Usecase	Menghapus data karyawan
Kode	SRS_F-SIPAR-009
Actor	Administrator
Deskripsi	Administrator dapat melakukan penghapusan pada data karyawan dalam sistem aplikasi SIPAR
Pre-kondisi	Administrator telah login ke sistem serta data karyawan telah sebelumnya tersimpan ke dalam sistem
Main Flow	1. Administrator mengakses website SIPAR 2. Administrator mengklik menu data karyawan 3. Administrator memilih opsi menghapus data karyawan 4. Administrator menghapus data karyawan sesuai dengan kebutuhannya
Alternative flow	Jika Administrator menekan “Cancel” maka proses akan dibatalkan dan tampilan akan kembali ke menu karyawan
Post-kondisi	Perubahan berupa penghapusan data karyawan telah tersimpan dalam sistem

Tabel 4.12 Skenario *use case* Melihat data rekapan keuangan

Nama Usecase	Melihat data rekapan keuangan
Kode	SRS_F-SIPAR-010
Actor	Administrator
Deskripsi	Administrator dapat melihat data rekapan keuangan

Pre-kondisi	Administrator telah login ke sistem serta data keuangan sebelumnya telah tersimpan ke dalam sistem
Main Flow	1. Administrator mengakses website SIPAR 2. Administrator mengklik menu data keuangan 3. Administrator memilih opsi melihat data rekapitulasi keuangan 4. Administrator melihat data rekapitulasi keuangan
Alternative flow	Jika pada SIPAR belum ada data rekapitulasi keuangan maka data yang ditampilkan adalah pernyataan bahwa data rekapitulasi keuangan tidak ditemukan.
Post-kondisi	Akan ditampilkan sebuah rekapitulasi data keuangan berdasarkan data-data keuangan yang telah disimpan dalam sistem

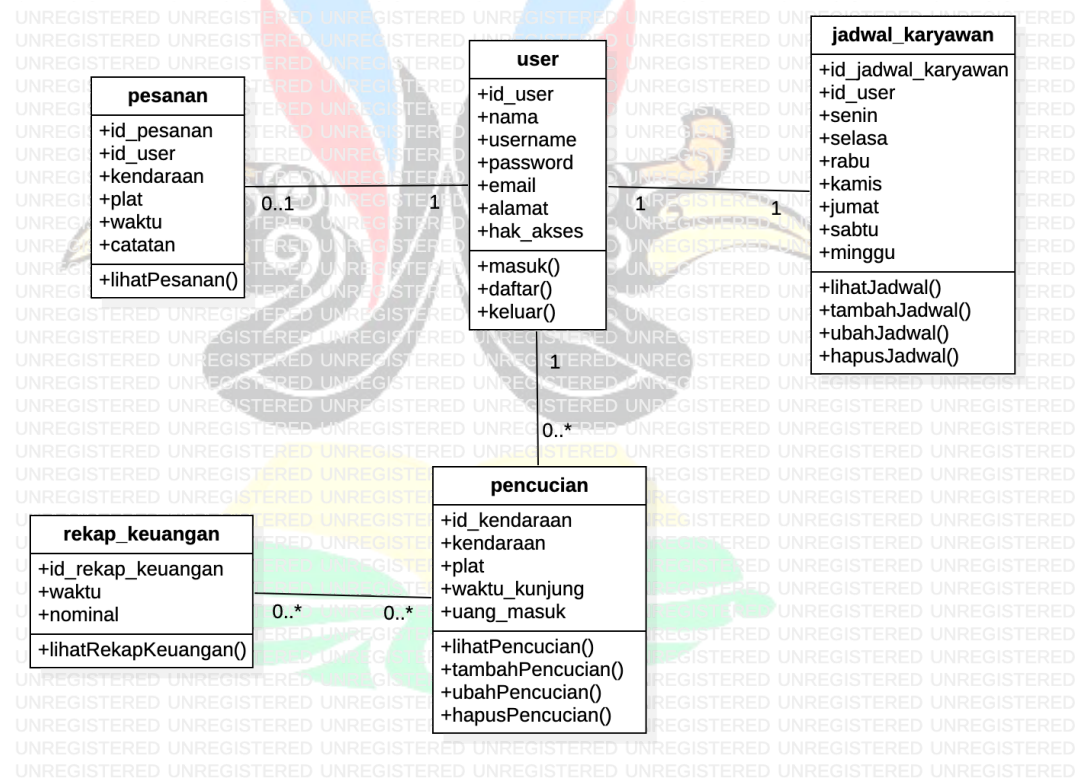
Tabel 4.13 Skenario use case Memasukkan data transaksi

Nama Usecase	Memasukkan data transaksi
Kode	SRS_F-SIPAR-013
Actor	Karyawan
Deskripsi	Karyawan dapat memasukkan data hasil transaksi ke dalam sistem aplikasi SIPAR
Pre-kondisi	Karyawan telah terdaftar dalam sistem serta karyawan telah melakukan login
Main Flow	1. Karyawan mengakses website SIPAR 2. Karyawan mengklik menu data keuangan 3. Karyawan memilih opsi memasukkan data hasil transaksi

	4. Karyawan memasukkan data hasil transaksi sesuai dengan keinginan dan/atau kebutuhannya
Alternative flow	Jika karyawan menekan “Cancel” maka proses akan dibatalkan dan tampilan akan kembali ke menu keuangan
Post-kondisi	Aplikasi akan menyimpan data hasil transaksi yang telah dimasukkan

4.3.6 Class Diagram

Diagram kelas (*class diagram*) menggambarkan kelas-kelas dalam sebuah sistem dan hubungannya antara satu dengan yang lain, serta dimasukkan juga atribut dan operasi



Gambar 4.2 Class Diagram Pencucian A&R

Berdasarkan Gambar 4.2 *Class Diagram* Pencucian A&R menggambarkan keterkaitan antar class dalam sistem. Class User berperan sebagai pusat

sistem dan berhubungan dengan beberapa class lainnya. Setiap User dapat memiliki satu atau tidak memiliki Pesanan, yang mencatat detail kendaraan, waktu, dan catatan tambahan. Selain itu, User juga terhubung dengan class Pencucian melalui relasi one-to-many, memungkinkan satu user memiliki banyak riwayat pencucian atau tidak sama sekali. Class Pencucian memuat informasi kendaraan, plat nomor, waktu layanan, dan pemasukan keuangan. Setiap aktivitas pencucian dapat dikaitkan dengan beberapa data pada class Rekap_Keuangan yang mencatat waktu dan nominal transaksi, membentuk relasi one-to-many. Di sisi lain, User memiliki relasi one-to-one dengan class Jadwal_Karyawan yang merekam jadwal kerja karyawan. Setiap *class* dilengkapi dengan metode untuk menampilkan, menambah, mengubah, atau menghapus data. Keseluruhan hubungan ini membentuk sistem yang saling terintegrasi antara pengguna, layanan pencucian, keuangan, dan manajemen jadwal.

4.4 System Design

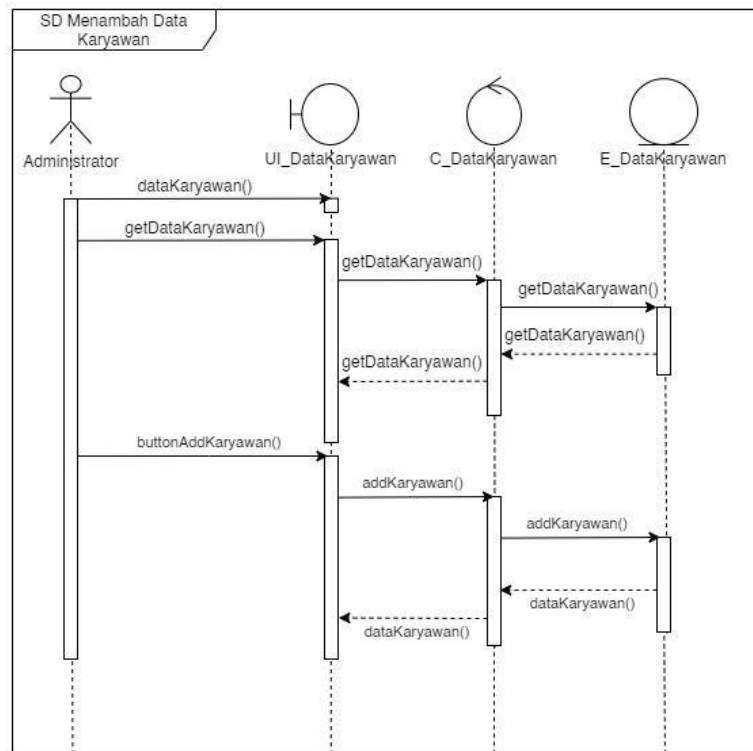
Tahap System Design merupakan fase di mana ditentukan fitur-fitur yang akan dikembangkan, dengan mempertimbangkan kebutuhan dan masukan dari para stakeholder. Tahapan ini berperan sebagai usulan solusi terhadap permasalahan yang telah diidentifikasi sebelumnya. Pada bagian ini dijelaskan rancangan antarmuka dari beberapa fitur utama maupun fitur pendukung yang terdapat dalam Sistem Informasi Pencucian A&R (SIPAR).

4.4.1 Deskripsi Arsitektural

Deskripsi arsitektural menjabarkan representasi alur hubungan antara pengguna dan sistem SIPAR, beserta keterkaitan antar modul yang terdapat di dalamnya. Rancangan sequence diagram berikut menggambarkan proses interaksi dan hubungan tersebut lebih rinci.

4.4.1.1 Menambah Data Karyawan

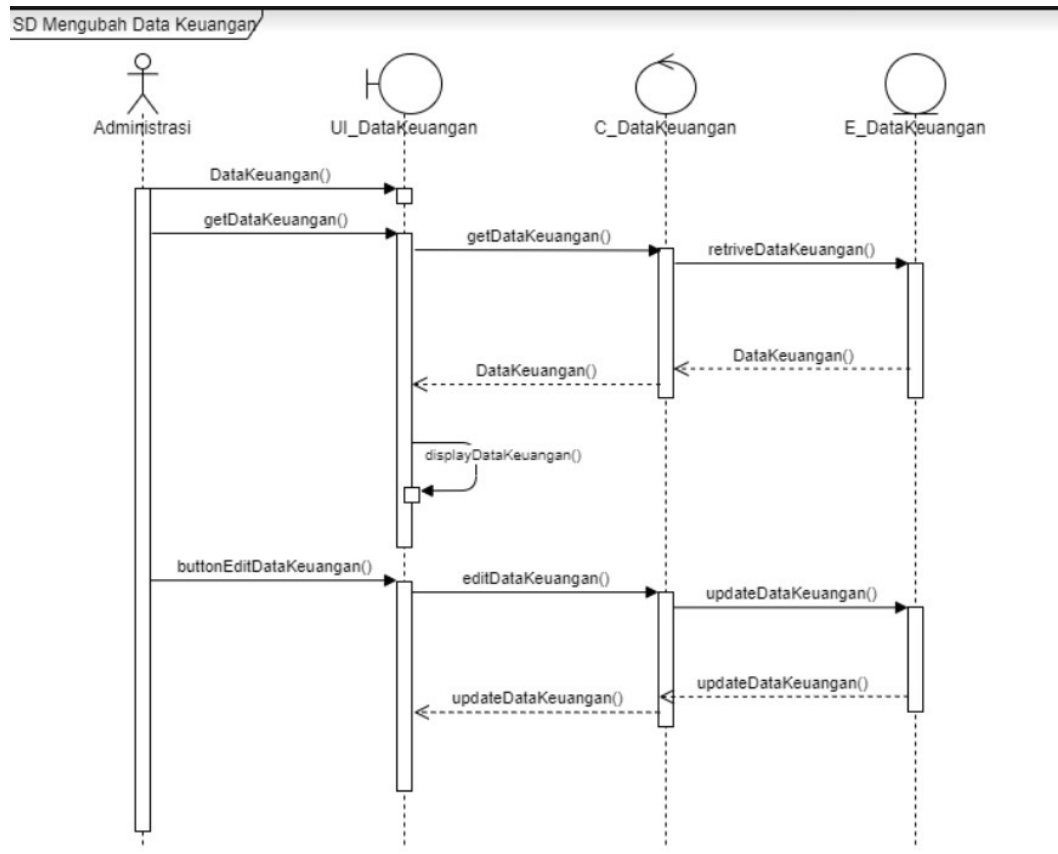
Fitur menambahkan data karyawan memberikan layanan kepada pengguna untuk dapat melakukan penambahan data karyawan yang baru. Data karyawan yang dapat ditambahkan meliputi Nama Lengkap, Username, Password, Email, Nomor Telepon, dan Alamat.



Gambar 4.3 Menambahkan Data Karyawan

4.4.1.2 Mengubah Data Keuangan

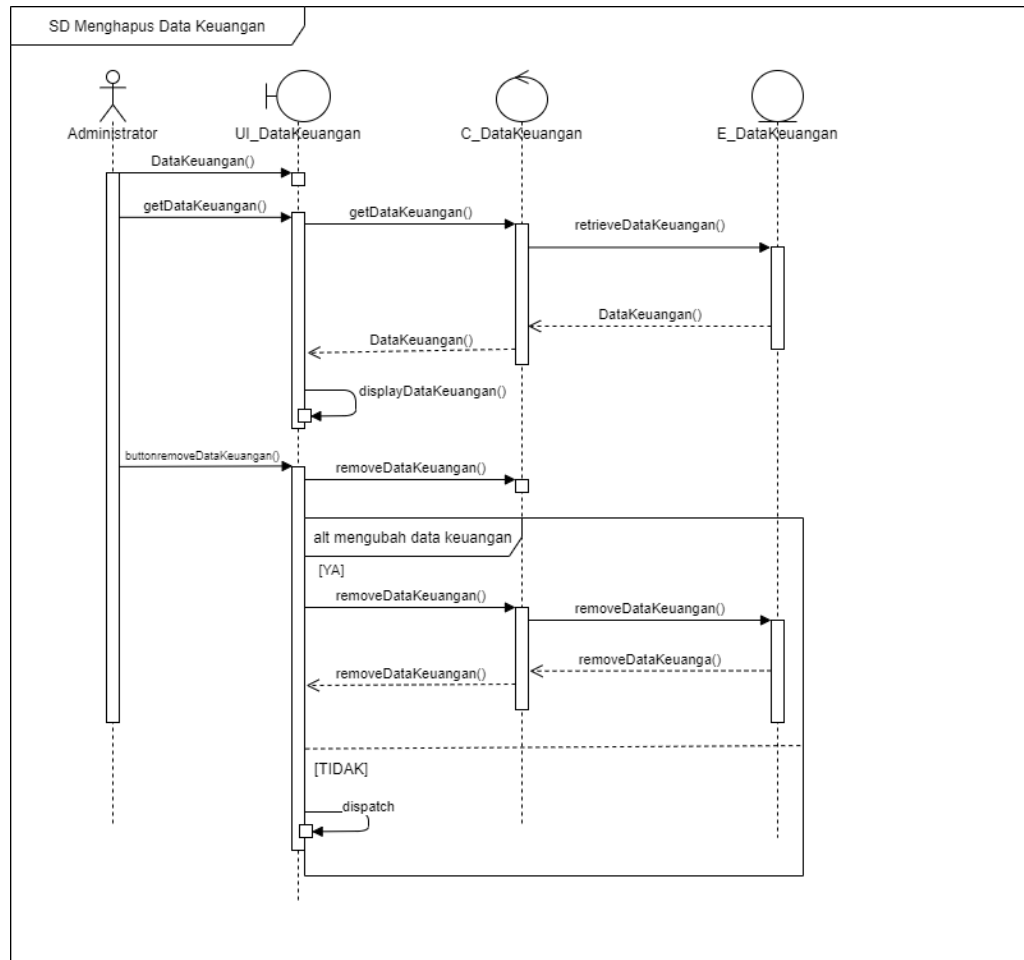
Fitur Mengubah Data Keuangan memberikan layanan kepada pengguna untuk dapat melakukan perubahan terhadap data keuangan yang telah diinput. Data yang dapat diubah pada fitur ini adalah data Jumlah uang transaksi dan data kendaraan.



Gambar 4.4 Mengubah Data Keuangan

4.4.1.3 Menghapus Data Keuangan

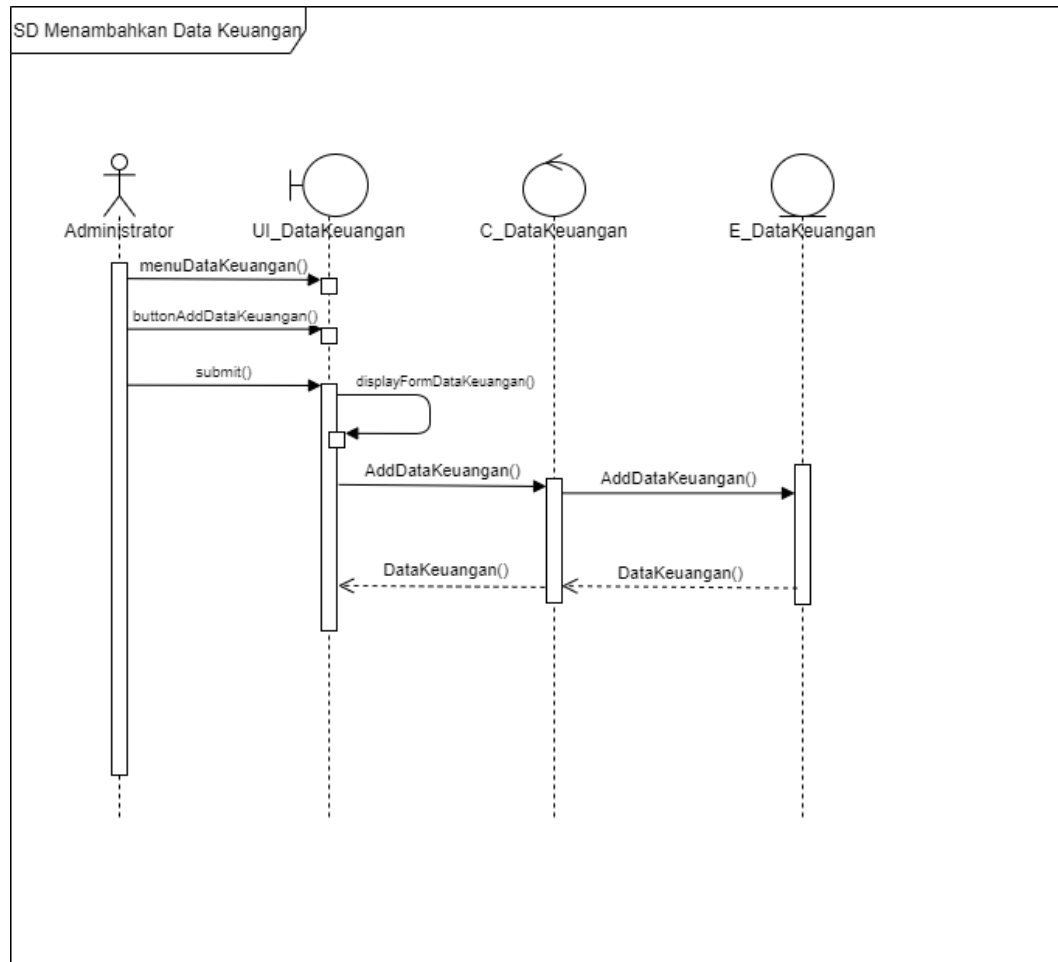
Fitur menghapus data keuangan memberikan layanan kepada pengguna untuk dapat menghapus data keuangan yang sebelumnya telah diinput. Fitur ini hanya dapat diakses oleh pengguna dengan akses *admin*.



Gambar 4.5 Menghapus Data Keuangan

4.4.1.4 Menambah Data Keuangan

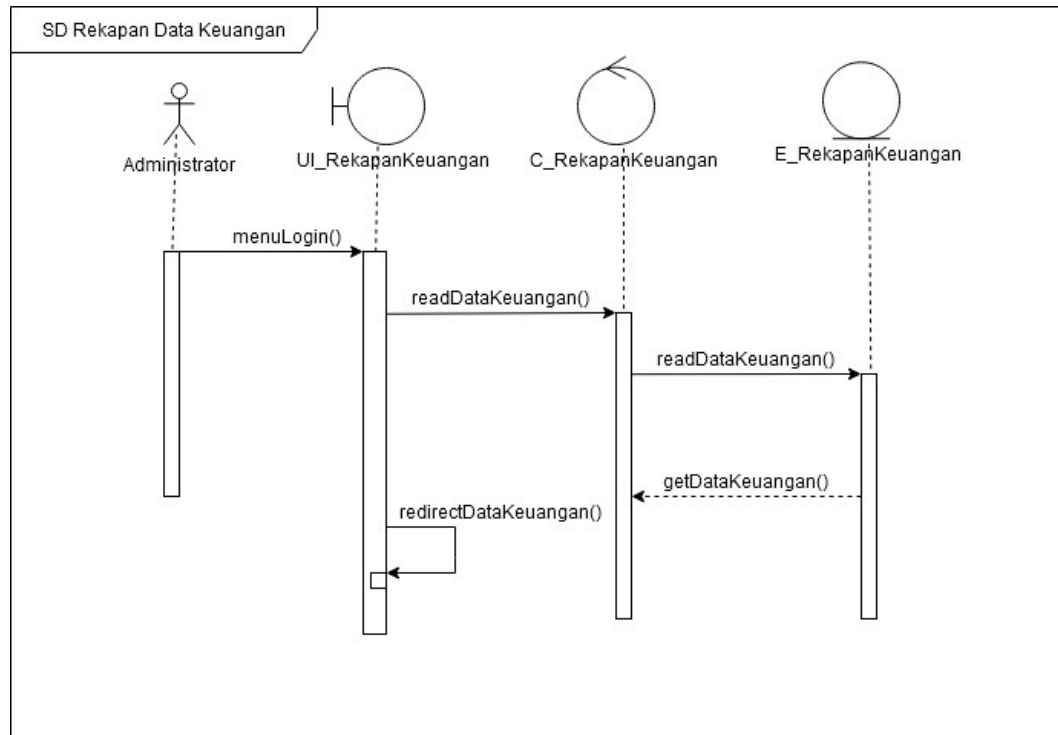
Fitur menambahkan data keuangan memberikan layanan kepada pengguna untuk menambahkan data keuangan berupa data transaksi. Data ini akan diakses oleh admin. Kemudian data akan tersimpan dan ditampilkan pada SIPAR di halaman laporan keuangan.



Gambar 4.6 Menambahkan Data Keuangan

4.4.1.5 Melihat Data Rekapitulasi Keuangan

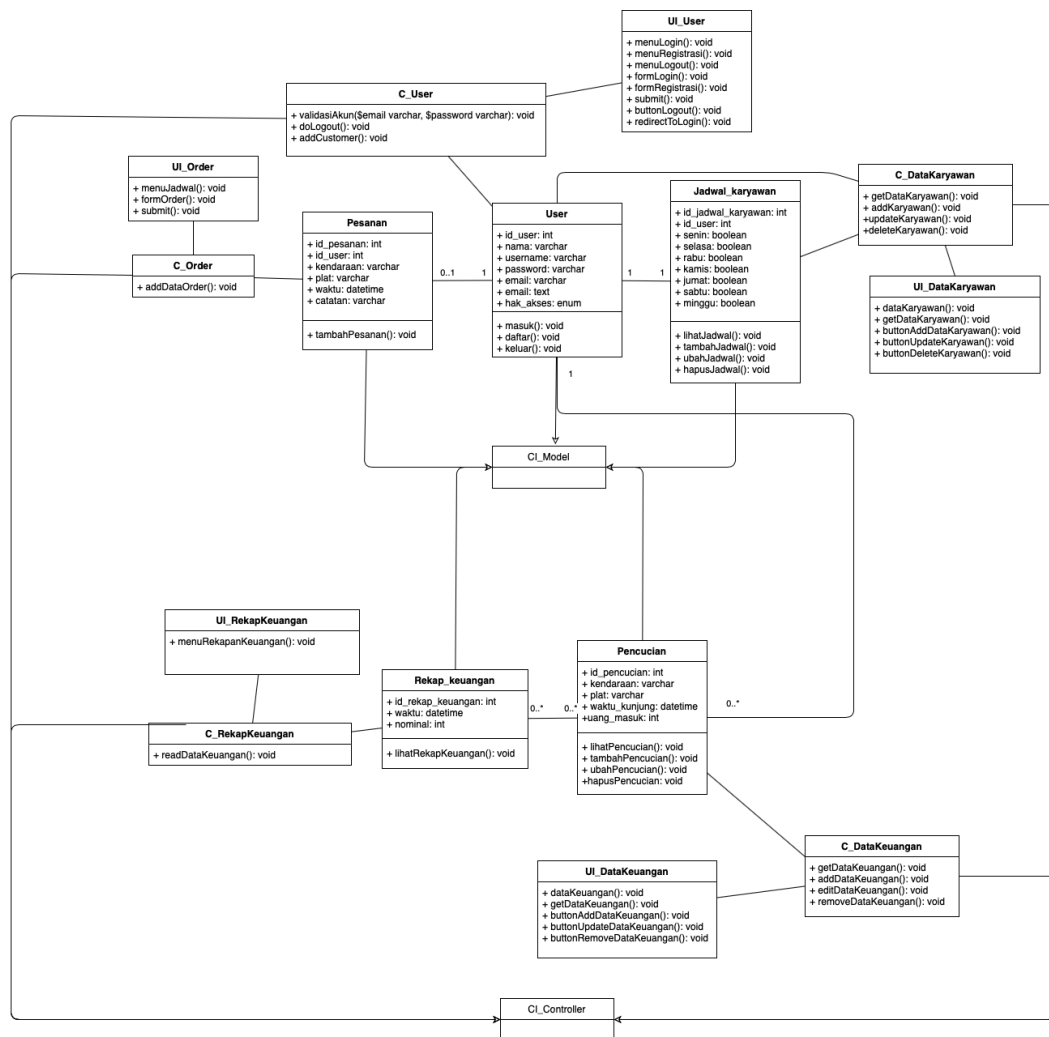
Fitur melihat data rekapitulasi keuangan memberikan layanan kepada pengguna untuk melihat hasil rekapitulasi data transaksi yang telah dilakukan dalam jangka waktu tertentu.



Gambar 4.7 Melihat Rekap Data Keuangan

4.4.2 Diagram Class Keseluruhan

Adapun penjelasan dari perubahan class diagram lebih detail dengan adanya penambahan class Model, View, dan Controller di masing-masing class. Salah satu contohnya yaitu class user terdapat class model user, class UI_User dan C_User. Pada semua class Controller akan terhubung ke class CI_Controller dan semua class model akan terhubung ke CI_Model



Gambar 4.8 Diagram Class Keseluruhan

4.5 Implementation

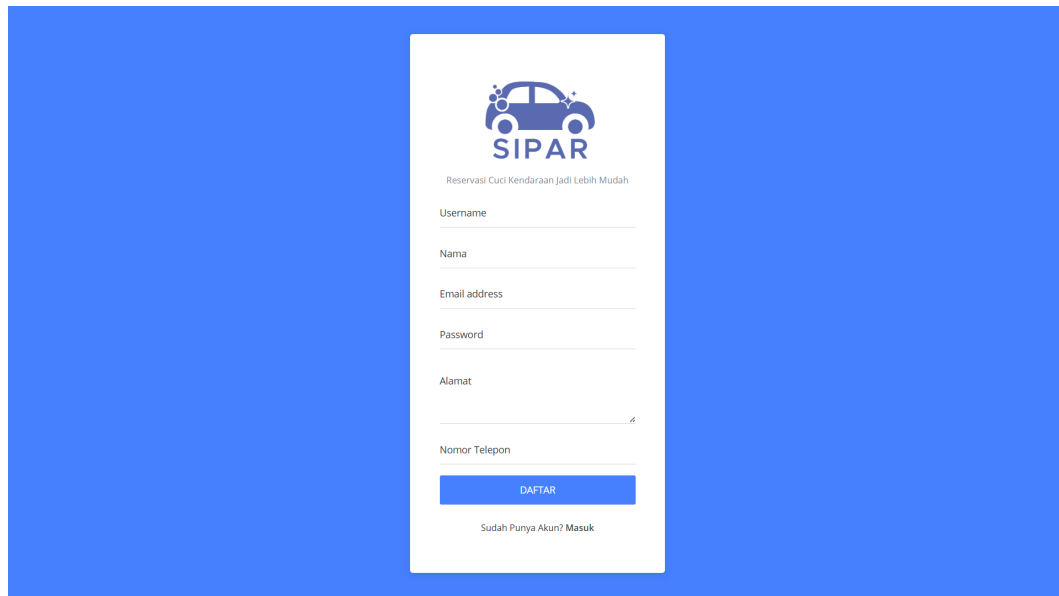
Tahap implementasi merupakan fase di mana proses penulisan kode program sistem dilakukan, berdasarkan acuan dari dokumen Deskripsi Perancangan Perangkat Lunak (DPPL). Dokumen ini berfungsi sebagai panduan teknis dalam pembangunan aplikasi SIPAR dan disertakan sebagai dokumen pendukung pada Lampiran C. Dalam proses pengembangan dan penggunaan SIPAR, koneksi ke jaringan internet menjadi prasyarat utama. Adapun *lingkup operasi* merujuk pada lingkungan operasional yang mencakup perangkat server dan client yang digunakan. Lingkungan operasi yang diterapkan pada sistem SIPAR adalah sebagai berikut:

4.5.1 Perancangan Antarmuka Register(SQ_001)

4.5.1.1 Spesifikasi fungsi *Register*

Identifikasi nama : *Register*

Deskripsi inti : Fungsi ini digunakan untuk melakukan pendaftaran akun agar pengguna dapat mengakses sistem



Gambar 4.9 Tampilan Halaman Register

4.5.1.2 Spesifikasi Algoritma

Tabel 4.14 Tabel Algoritma Register

No.	Kode
1	public function register()
2	{
3	if (empty(\$this->input->post('nama')))) {
4	\$this->load->view('auth/template/header');
5	\$this->load->view('auth/signup');
6	\$this->load->view('auth/template/footer');
7	} else {
8	\$data = array(
9	'username' => \$this->input->post('username'),
10	'email' => \$this->input->post('email'),
11	'password' => md5(\$this->input->post('password')),

12	'nama' => \$this->input->post('nama'),
13	'alamat' => \$this->input->post('alamat'),
14	'nohp' => \$this->input->post('nohp'),
15	'hak_akses' => 'pelanggan'
16	);
17	
18	\$this->UserModel->addUser(\$data);
19	\$this->session->set_flashdata('pesan', '<div class="alert
20	alert-success alert-dismissible fade show" role="alert">
21	Selamat! Anda berhasil mendaftar, silahkan masukkan username dan
22	password anda untuk masuk.
23	<button type="button" class="close" data-dismiss="alert"
24	aria-label="Close">
25	×</button>
26	</div>');
27	
28	redirect(base_url('login'));
29	}
30	}

Berdasarkan Tabel 4.14 ditampilkan implementasi kode program untuk halaman *Register* yang berfungsi mengatur fungsionalitas proses pendaftaran akun pengguna. Penjelasan terkait struktur dan fungsi dari setiap bagian kode tersebut dijabarkan pada Tabel 4.15 sebagai berikut:

Tabel 4.15 Penjelasan Kode Tampilan *Register*

Baris Kode	Deskripsi Kode
1-2	Fungsi ini digunakan untuk mengatur proses pendaftaran akun baru oleh pengguna.
3-6	Kalau pengguna belum mengisi form (belum klik daftar), maka sistem akan menampilkan halaman pendaftaran
7-16	Kalau pengguna sudah mengisi dan mengirim form, sistem akan mengambil data dari isian seperti nama, username, email, password, alamat, dan nomor HP, lalu memasukkannya ke dalam satu paket data.
17	Data pendaftaran tersebut akan dikirim ke database lewat UserModel, supaya bisa disimpan.
18-23	Setelah data tersimpan, sistem menampilkan pesan bahwa

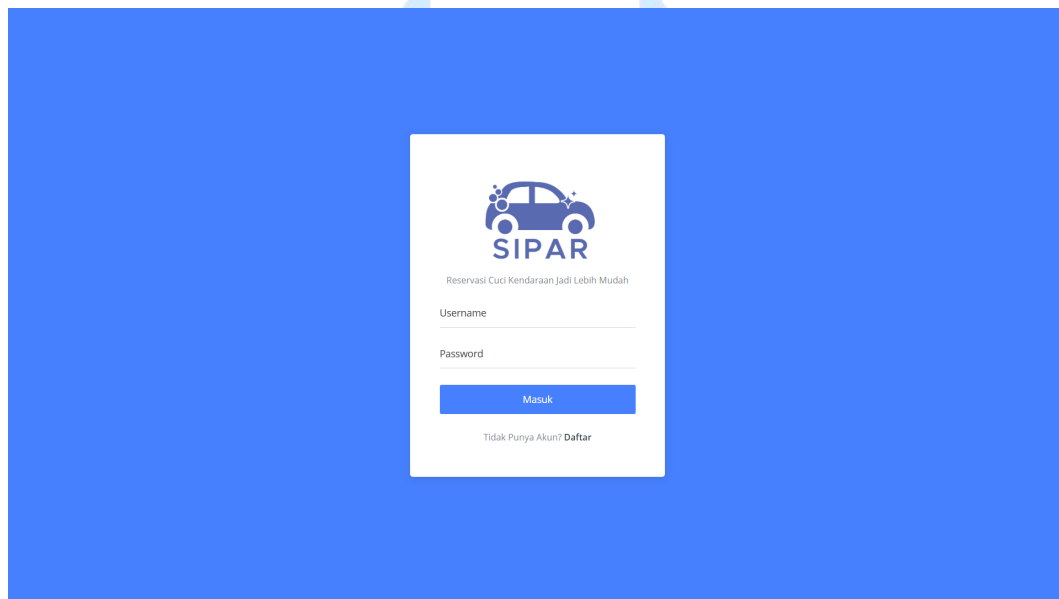
	pendaftaran berhasil dan pengguna bisa login.
24	Setelah itu, pengguna otomatis diarahkan ke halaman login.

4.5.2 Perancangan Antarmuka Login (SQ_002)

4.5.2.1 Spesifikasi fungsi *Login*

Identifikasi nama : *Login*

Deskripsi inti : Fungsi ini digunakan untuk melakukan verifikasi terhadap username dan password yang dimasukkan oleh pengguna.



Gambar 4.10 Login

4.5.2.2 Spesifikasi Algoritma

Tabel 4.16 Tabel Algoritma Login

No.	Kode
1	class Auth extends CI_Controller
2	{
3	public function __construct()
4	{
5	parent::__construct();
6	\$this->load->model('UserModel');
7	}

8	
9	public function index()
10	{
11	if (empty(\$this->input->post('username'))){
12	\$this->load->view('auth/template/header');
13	\$this->load->view('auth/login');
14	\$this->load->view('auth/template/footer');
15	} else {
16	\$where = array(
17	'username' => \$this->input->post('username'),
18	'password' => md5(\$this->input->post('password'))
19	);
20	\$user = \$this->UserModel->cekUser(\$where);
21	echo \$user->num_rows();
22	
23	if (\$user->num_rows() > 0) {
24	\$data = \$user->row();
25	\$data_session = array(
26	'id_user' => \$data->id_user,
27	'username' => \$data->username,
28	'nama' => \$data->nama,
29	'hak_akses' => \$data->hak_akses
30	);
31	
32	\$this->session->set_userdata(\$data_session);
33	
34	if (\$data->hak_akses == 'pelanggan') {
35	redirect('/dashboard');
36	} else if (\$data->hak_akses == 'admin') {
37	redirect('/admin/dashboard');
38	} else {
39	redirect('/karyawan/dashboard');
40	}
41	} else {
42	\$this->session->set_flashdata('pesan', '<div
43	class="alert alert-danger alert-dismissible fade show" role="alert">
44	Username dan Password yang anda masukkan tidak
45	ditemukan.
46	<button type="button" class="close" data-dismiss="alert"
47	aria-label="Close">
48	×
49	</button>
50	</div>');
51	redirect(base_url('login'));
52	}
53	}
54	}

55	
56	
57	
58	

Berdasarkan Tabel 4.16 ditampilkan implementasi kode program untuk halaman *Login* yang berfungsi mengatur tampilan serta fungsionalitas proses autentikasi pengguna. Penjelasan terkait struktur dan fungsi dari setiap bagian kode tersebut dijabarkan pada Tabel 4.16 sebagai berikut:

Tabel 4.17 Penjelasan Kode Tampilan *Login*

Baris Kode	Deskripsi Kode
1	Mendefinisikan Auth sebagai controller yang mewarisi CI_Controller, bertanggung jawab atas proses autentikasi pengguna.
3-7	Konstruktor yang memuat UserModel untuk memungkinkan pengambilan dan pengecekan data pengguna dari basis data.
8-9	Fungsi utama controller ini. Akan dipanggil ketika pengguna mengakses
10-13	Jika form belum dikirim (input username kosong), tampilkan halaman login dengan memuat tiga view: header, login, dan footer.
14-18	Jika form dikirim, ambil username dan password, lalu hash password dengan md5 untuk mencocokkan dengan database. Data disiapkan dalam array \$where
19-20	Menggunakan model UserModel untuk mengecek keberadaan user dengan data login tersebut. echo digunakan hanya untuk debugging (menampilkan jumlah data yang ditemukan).
21-22	Jika data user ditemukan (jumlah baris hasil query > 0), ambil satu baris data sebagai objek.
23-29	Menyusun data user untuk disimpan dalam session, lalu menyimpannya menggunakan set_userdata. Data ini akan digunakan selama sesi login berlangsung.
30-35	Melakukan pengecekan hak akses dan mengarahkan user ke dashboard yang sesuai: pelanggan, admin, atau karyawan.

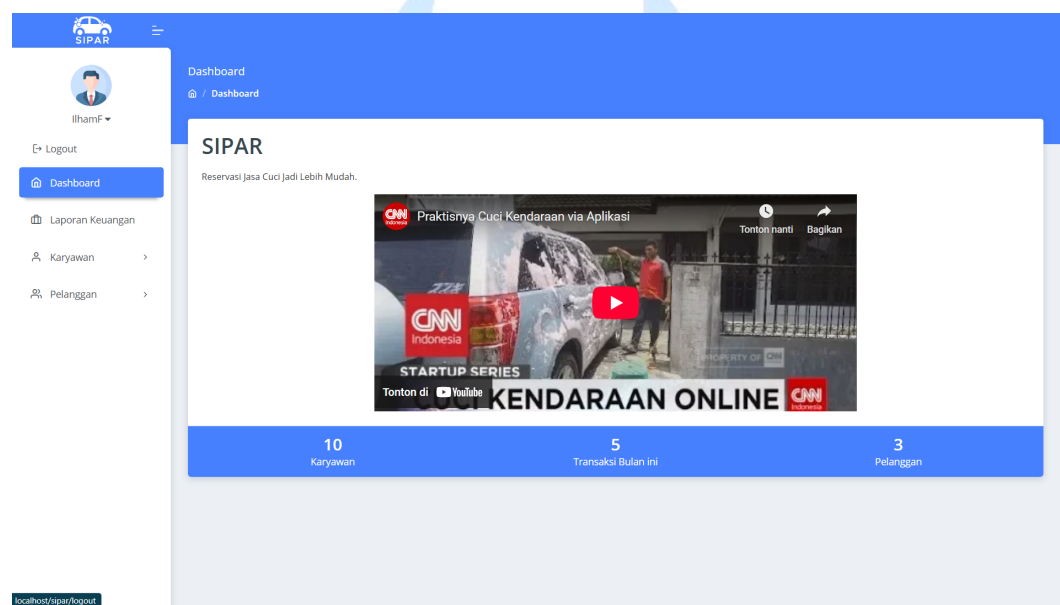
37-43	Jika user tidak ditemukan (login gagal), simpan pesan kesalahan ke session dan kembalikan user ke halaman login.
-------	--

4.5.3 Perancangan *Logout* (SQ_003)

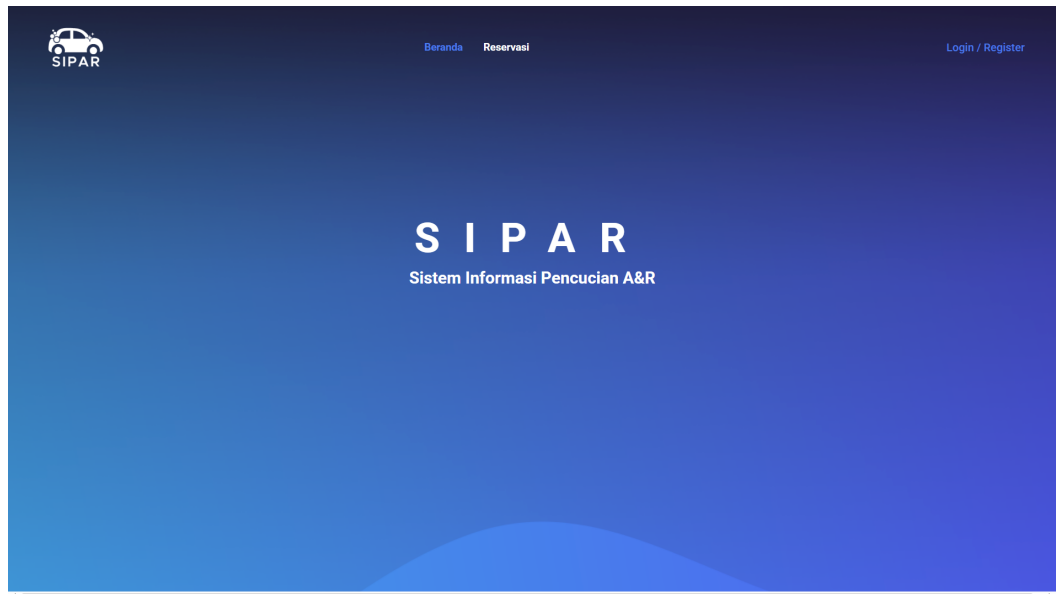
4.5.3.1 Spesifikasi fungsi *Logout*

Identifikasi nama : *Login*

Deskripsi inti : Fungsi ini digunakan pengguna untuk keluar dari sistem.



Gambar 4.11 Logout (1)



Gambar 4.12 Logout (2)

4.5.3.2 Spesifikasi Algoritma

Tabel 4.18 Tabel Algoritma Logout

No.	Kode
1	public function logout()
2	{
3	\$this->session->sess_destroy();
4	redirect('/');
5	}
6	}

Berdasarkan Tabel 4.18 ditampilkan implementasi kode program untuk halaman *Logout* yang berfungsi mengatur fungsionalitas proses pengguna keluar dari sistem. Penjelasan terkait struktur dan fungsi dari setiap bagian kode tersebut dijabarkan pada Tabel 4.19 sebagai berikut:

Tabel 4.19 Penjelasan Kode Tampilan *Logout*

Baris Kode	Deskripsi Kode
1-2	Method logout() digunakan untuk mengakhiri sesi pengguna yang sedang login.

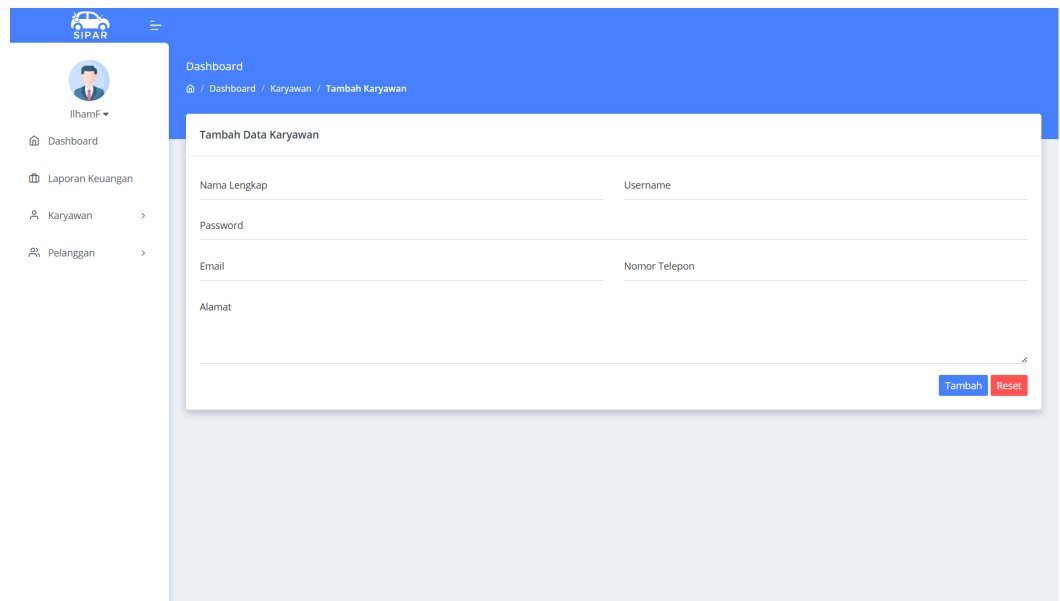
2-4	Semua data session pengguna akan dihapus menggunakan sess_destroy(), kemudian sistem mengarahkan pengguna ke halaman utama atau beranda
-----	---

4.5.4 Perancangan Antarmuka Menambah Data Karyawan (SQ_004)

4.5.1.1 Spesifikasi fungsi Menambah Data Karyawan

Identifikasi nama : Menambah Data Karyawan

Deskripsi inti : Fungsi ini digunakan untuk menambahkan Data Karyawan



Gambar 4.13 Menambah Data Karyawan

4.5.1.2 Spesifikasi Algoritma

Tabel 4.20 Tabel Algoritma Menambah Data Karyawan

No.	Kode
1	public function addKaryawan()
2	{
3	if (empty(\$this->input->post('nama'))) {
4	\$this->load->view('admin/template/header');
5	\$this->load->view('admin/tambah-karyawan');
6	\$this->load->view('admin/template/footer');
7	} else {
8	\$data = array(

9	'username' => \$this->input->post('username'),
10	'email' => \$this->input->post('email'),
11	'password' => md5(\$this->input->post('password')),
12	'nama' => \$this->input->post('nama'),
13	'alamat' => \$this->input->post('alamat'),
14	'nohp' => \$this->input->post('nohp'),
15	'hak_akses' => 'karyawan'
16	);
17	
18	
19	\$this->UserModel->addUser(\$data);
20	\$this->session->set_flashdata('pesan', '<div class="alert alert-success
21	alert-dismissible fade show" role="alert">
22	Data Karyawan Berhasil Ditambahkan!
23	<button type="button" class="close" data-dismiss="alert"
24	aria-label="Close">
25	×</button>
26	</div>');
27	redirect('admin/karyawan');
28	}
29	}
30	}

Berdasarkan Tabel 4.20 ditampilkan implementasi kode program untuk halaman Menambahkan Data Karyawan yang berfungsi mengatur tampilan serta fungsionalitas proses menambahkan data karyawan yang dilakukan oleh pengguna. Penjelasan terkait struktur dan fungsi dari setiap bagian kode tersebut dijabarkan pada Tabel 4.21 sebagai berikut:

Tabel 4.21 Penjelasan Kode Tampilan Menambah Karyawan

Baris Kode	Deskripsi Kode
1-2	fungsi untuk menambahkan data karyawan baru ke sistem, biasanya dilakukan oleh admin.
3-6	Kalau form belum diisi (nama masih kosong), sistem akan menampilkan halaman tambah karyawan agar admin bisa mengisi data.
7-15	Kalau form sudah dikirim, maka semua data karyawan seperti username, email, password, nama, alamat, dan no HP akan diambil dari form dan dikumpulkan.

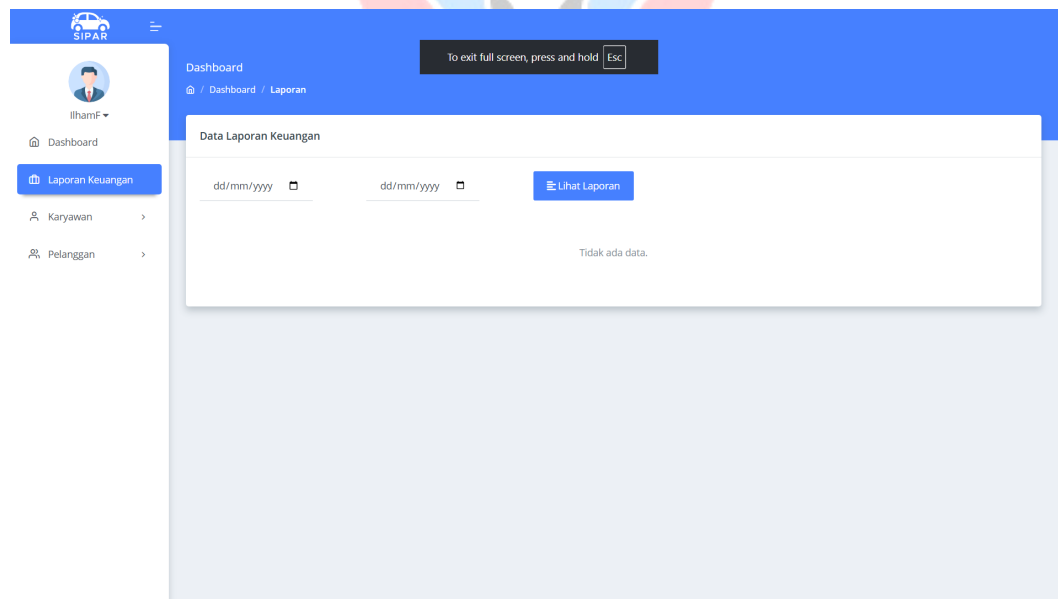
16	Sistem langsung menetapkan bahwa akun ini adalah akun karyawan (bukan admin atau pelanggan).
18	Data tersebut dikirim dan disimpan ke database lewat UserModel
19-24	Setelah berhasil disimpan, sistem menampilkan pesan sukses agar admin tahu data karyawan sudah masuk.
25	Setelah selesai, admin akan langsung dibawa ke halaman daftar karyawan.

4.5.5 Perancangan Antarmuka Mengubah Data Keuangan (SQ_006)

4.5.1.1 Spesifikasi fungsi Menghapus Data Keuangan

Identifikasi nama : Mengubah Data Keuangan

Deskripsi inti : Fungsi ini digunakan untuk melakukan penghapusan terhadap data keuangan yang telah dimasukkan sebelumnya.



Gambar 4.14 Menghapus Data Keuangan

4.5.1.2 Spesifikasi Algoritma

Tabel 4.22 Tabel Algoritma Menghapus Data Keuangan

No.	Kode
1	public function deleteReservasi(\$id)
2	{
3	\$this->ReservasiModel->deleteReservasi(\$id);
4	\$this->session->set_flashdata('pesan', '<div class="alert alert-danger
5	alert-dismissible fade show" role="alert">
6	Data Reservasi Berhasil Dibatalkan!
7	<button type="button" class="close" data-dismiss="alert"
8	aria-label="Close">
9	×
10	</button>
11	</div>');
12	redirect('admin/reservasi');
13	}
14	}

Berdasarkan Tabel 4.22 ditampilkan implementasi kode program untuk halaman Menghapus Data Keuangan yang berfungsi mengatur tampilan serta fungsionalitas proses Menghapus data yang dilakukan oleh pengguna. Penjelasan terkait struktur dan fungsi dari setiap bagian kode tersebut dijabarkan pada Tabel 4.22 sebagai berikut:

Tabel 4.23 Penjelasan Kode Tampilan Menghapus Data Keuangan

Baris Kode	Deskripsi Kode
1-2	Fungsi ini digunakan untuk menghapus (membatalkan) data reservasi berdasarkan ID tertentu.
3	Menghapus data reservasi dari database dengan memanggil fungsi deleteReservasi() di model ReservasiModel, menggunakan ID sebagai acuannya.
4-9	Setelah data berhasil dihapus, sistem akan menyimpan pesan notifikasi sementara (flashdata) yang memberi tahu bahwa reservasi berhasil dibatalkan.
10	Setelah proses selesai, admin akan diarahkan kembali ke halaman daftar reservasi.

4.5.6 Perancangan Antarmuka Menambahkan Data Keuangan

4.5.6.1 Spesifikasi fungsi Menambahkan Data Keuangan

Identifikasi nama : Menambahkan Data Keuangan
 Deskripsi inti : Fungsi ini digunakan untuk Menambahkan data keuangan berupa transaksi.

Gambar 4.15 Menambahkan Data Keuangan

4.5.6.2 Spesifikasi Algoritma

Tabel 4.24 Tabel Algoritma Menambahkan Data Keuangan

No.	Kode
1	public function addReservasi()
2	{
3	if (empty(\$this->input->post('kendaraan'))){
4	\$data['pelanggan'] = \$this->PelangganModel->getPelanggan();
5	\$this->load->view('admin/template/header');
6	\$this->load->view('admin/tambah-reservasi', \$data);
7	\$this->load->view('admin/template/footer');
8	} else {
9	\$data = array(
10	'id_user' => \$this->input->post('user'),
11	'kendaraan' => \$this->input->post('kendaraan'),
12	'plat' => \$this->input->post('plat'),
13	'uang_masuk' => \$this->input->post('uang_masuk'),
14	'status' => 'selesai'
15	);
16	}

17	
18	<code>\$this->ReservasiModel->addPencucian(\$data);</code>
19	<code>\$this->session->set_flashdata('pesan', '<div class="alert alert-success</code>
20	<code>alert-dismissible fade show" role="alert"></code>
21	<code> Data Pencucian Berhasil Ditambahkan!</code>
22	<code> <button type="button" class="close" data-dismiss="alert"</code>
23	<code>aria-label="Close"></code>
24	<code> &times;</code>
25	<code> </button></code>
26	<code></div>');</code>
27	<code> redirect('admin/reservasi');</code>
28	<code>}</code>
29	<code>}</code>

Berdasarkan Tabel 4.24 ditampilkan implementasi kode program untuk halaman Menambahkan data keuangan yang berfungsi mengatur tampilan serta fungsionalitas proses Menambahkan data keuangan yang dilakukan oleh pengguna. Penjelasan terkait struktur dan fungsi dari setiap bagian kode tersebut dijabarkan pada Tabel 4.25 sebagai berikut:

Tabel 4.25 Penjelasan Kode Tampilan *Login*

Baris Kode	Deskripsi Kode
1-2	Fungsi ini digunakan oleh admin untuk menambahkan data reservasi atau pencucian kendaraan.
3-6	Jika form reservasi belum diisi, maka sistem akan menampilkan halaman tambah reservasi, termasuk mengambil data pelanggan dari database.
4	Mengambil daftar pelanggan dari database agar bisa dipilih di form reservasi.
5-7	Menampilkan halaman form tambah reservasi beserta header dan footer-nya.
8-15	Jika form sudah diisi dan dikirim, sistem akan mengambil data dari form seperti: siapa pelanggannya (id_user), jenis kendaraan, plat nomor, jumlah uang yang dibayar, dan otomatis memberi status “selesai”.
16	Data tersebut dikirim ke model ReservasiModel untuk disimpan ke database sebagai data baru.

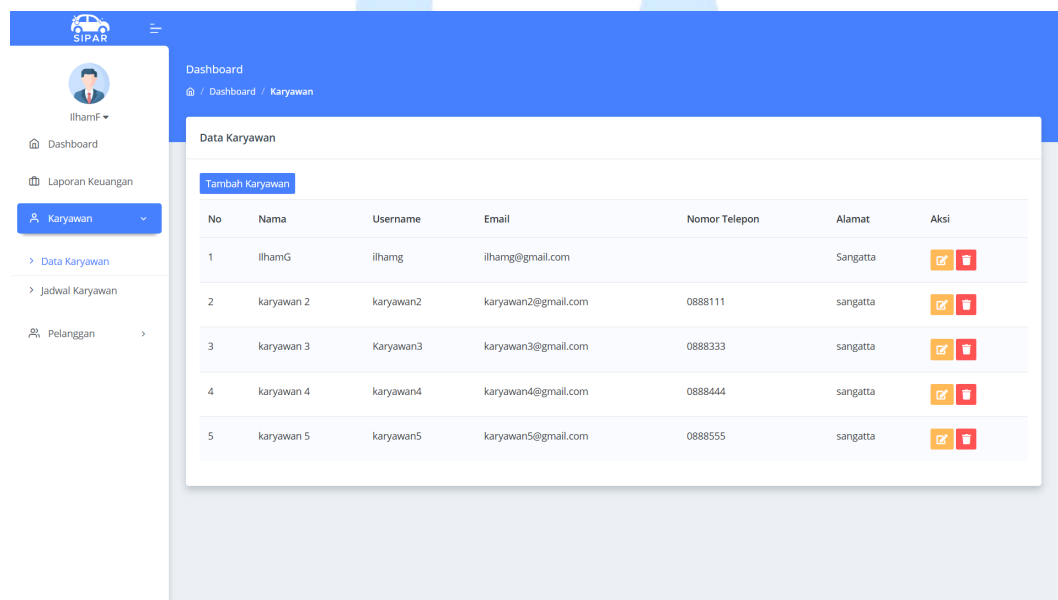
17-22	Setelah berhasil disimpan, sistem akan menampilkan pesan sukses bahwa data pencucian berhasil ditambahkan.
23	Setelah proses selesai, admin akan diarahkan kembali ke halaman daftar reservasi.

4.5.7 Perancangan Antarmuka Mengubah Data Karyawan (SQ_008)

4.5.1.1 Spesifikasi fungsi Mengubah Data Karyawan

Identifikasi nama : Mengubah Data Karyawan

Deskripsi inti : Fungsi ini digunakan untuk melakukan Perubahan terhadap data karyawan



Gambar 4.16 Mengubah Data Karyawan

4.5.1.2 Spesifikasi Algoritma

Tabel 4.26 Tabel Algoritma Mengubah Data Karyawan

No.	Kode
1	public function editKaryawan(\$id)
2	{
3	if (empty(\$this->input->post('nama'))){
4	\$data['karyawan'] = \$this->KaryawanModel->getByldKaryawan(\$id);

5	<code>\$this->load->view('admin/template/header');</code>
6	<code>\$this->load->view('admin/edit-karyawan', \$data);</code>
7	<code>\$this->load->view('admin/template/footer');</code>
8	<code>} else {</code>
9	<code> \$data = array(</code>
10	<code> 'username' => \$this->input->post('username'),</code>
11	<code> 'email' => \$this->input->post('email'),</code>
12	<code> 'nama' => \$this->input->post('nama'),</code>
13	<code> 'alamat' => \$this->input->post('alamat'),</code>
14	<code> 'nohp' => \$this->input->post('nohp'),</code>
15	<code> 'hak_akses' => 'karyawan'</code>
16	<code>);</code>
17	
18	<code> \$this->KaryawanModel->editKaryawan(\$data, \$id);</code>
19	<code> \$this->session->set_flashdata('pesan', '<div class="alert alert-success</code>
20	<code>alert-dismissible fade show" role="alert"></code>
21	<code> Data Karyawan Berhasil Diubah!</code>
22	<code> <button type="button" class="close" data-dismiss="alert"</code>
23	<code>aria-label="Close"></code>
24	<code> &times;</button></code>
25	<code> </div>');</code>
26	<code> redirect('admin/karyawan');</code>
27	<code>}</code>
28	<code>}</code>
29	<code>}</code>

Berdasarkan Tabel 4.27 ditampilkan implementasi kode program untuk halaman Mengubah Data Karyawan yang berfungsi mengatur tampilan serta fungsionalitas proses autentikasi pengguna. Penjelasan terkait struktur dan fungsi dari setiap bagian kode tersebut dijabarkan pada Tabel 4.27 sebagai berikut:

Tabel 4.27 Penjelasan Kode Tampilan Mengubah Data Karyawan

Baris Kode	Deskripsi Kode
1-2	Fungsi ini digunakan oleh admin untuk mengedit atau memperbarui data karyawan berdasarkan ID-nya.
3-6	Kalau form edit belum diisi (misalnya halaman baru dibuka), sistem akan mengambil data karyawan berdasarkan ID lalu menampilkan halaman form edi
4	Mengambil data karyawan dari database berdasarkan ID untuk ditampilkan di form.

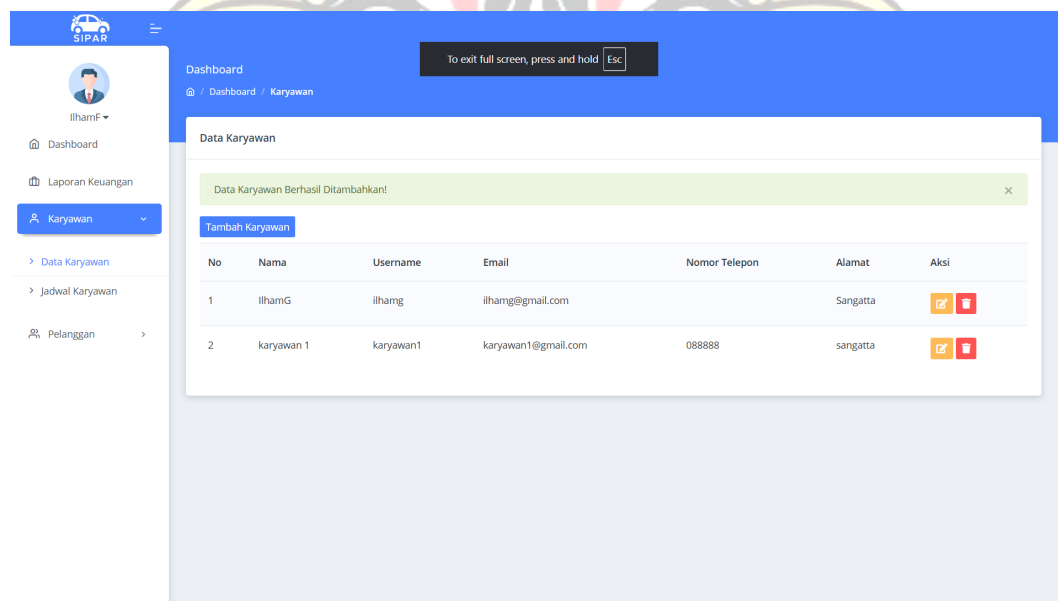
5-7	Menampilkan halaman form edit karyawan beserta header dan footer-nya.
8-15	Kalau form sudah diisi dan dikirim, sistem akan mengambil semua data yang baru dari input: username, email, nama, alamat, no HP, dan menetapkan peran sebagai karyawan.
16	Data baru tersebut dikirim ke model KaryawanModel untuk memperbarui data di database berdasarkan ID karyawan.
17-22	Sistem menyimpan pesan sukses untuk memberi tahu bahwa data karyawan berhasil diperbarui.
23	Setelah selesai, admin akan diarahkan ke halaman daftar karyawan.

4.5.8 Perancangan Antarmuka Menghapus Data Karyawan (SQ_009)

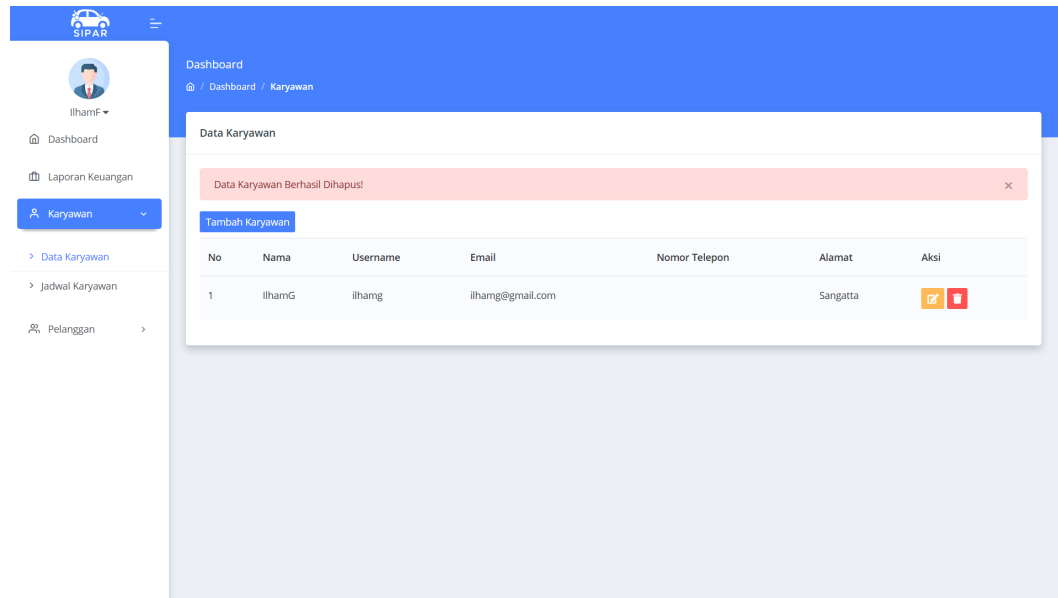
4.5.1.1 Spesifikasi fungsi Menghapus Data Karyawan

Identifikasi nama : Menghapus Data Karyawan

Deskripsi inti : Fungsi ini digunakan untuk melakukan proses menghapus data karyawan



Gambar 4.17 Menghapus Data Karyawan (1)



Gambar 4.17 Menghapus Data Karyawan (2)

4.5.1.2 Spesifikasi Algoritma

Tabel 4.28 Tabel Algoritma Menghapus Data Karyawan

No.	Kode
1	public function deleteKaryawan(\$id)
2	{
3	\$this->KaryawanModel->deleteKaryawan(\$id);
4	\$this->session->set_flashdata('pesan', '<div class="alert alert-danger
5	alert-dismissible fade show" role="alert">
6	Data Karyawan Berhasil Dihapus!
7	<button type="button" class="close" data-dismiss="alert"
8	aria-label="Close">
9	×
10	</button>
11	</div>');
12	redirect('admin/karyawan');
13	}

Berdasarkan Tabel 4.28 ditampilkan implementasi kode program untuk halaman menghapus data karyawan yang berfungsi mengatur tampilan serta fungsionalitas proses menghapus data karyawan pengguna. Penjelasan terkait struktur dan fungsi dari setiap bagian kode tersebut dijabarkan pada Tabel 4. sebagai berikut:

Tabel 4.29 Penjelasan Kode Tampilan menghapus data karyawan

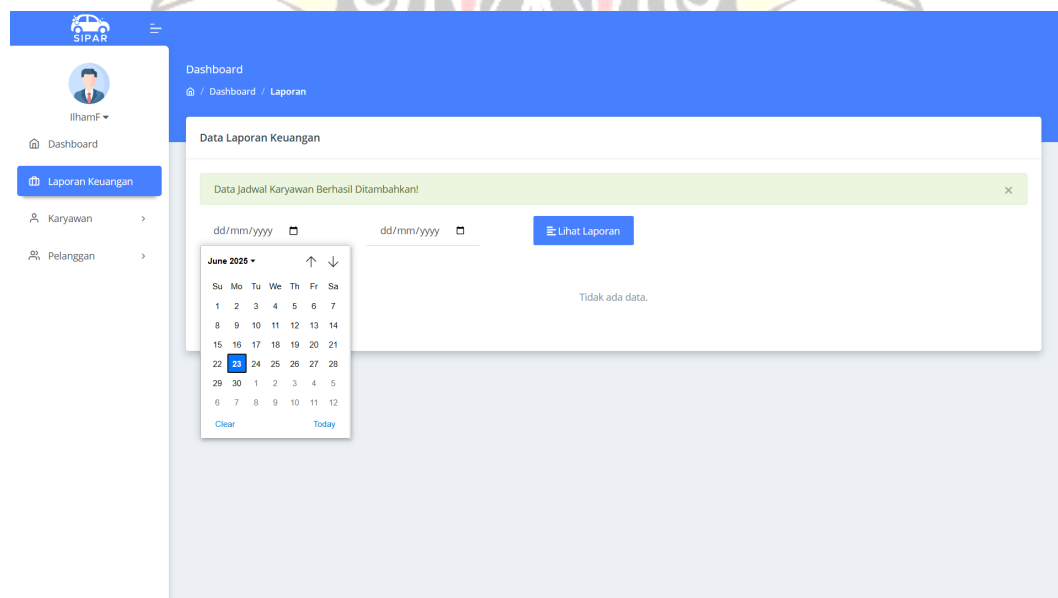
Baris Kode	Deskripsi Kode
1	Fungsi ini digunakan untuk menghapus data karyawan berdasarkan ID-nya.
3-7	Menghapus data karyawan dari database menggunakan fungsi di KaryawanModel, berdasarkan ID yang dikirim.
8-9	Setelah data dihapus, sistem menampilkan pesan pemberitahuan bahwa karyawan berhasil dihapus.
10-13	Setelah proses selesai, admin langsung diarahkan kembali ke halaman daftar karyawan.
37-43	

4.5.9 Perancangan Antarmuka Melihat Data Rekapitulasi Keuangan (SQ_010)

4.5.1.1 Spesifikasi fungsi Melihat Data Rekapitulasi Keuangan

Identifikasi nama : Melihat Data Rekapitulasi Keuangan

Deskripsi inti : Fungsi ini digunakan untuk menampilkan data rekapitulasi keuangan di sistem



Gambar 4.18 Melihat Data Rekapitulasi Keuangan

4.5.1.2 Spesifikasi Algoritma

Tabel 4.30 Tabel Algoritma Melihat Data Rekapitulasi Keuangan

No.	Kode
1	<?php
2	class Laporan extends CI_Controller
3	{
4	public function __construct()
5	{
6	parent::__construct();
7	
8	\$this->load->model('LaporanModel');
9	\$this->load->model('UserModel');
10	
11	if (\$this->session->userdata('nama') == NULL) {
12	redirect(base_url('login'));
13	}
14	}
15	public function index()
16	{
17	\$this->load->view('admin/template/header');
18	\$this->load->view('admin/laporan');
19	}
20	
21	public function getData()
22	{
23	\$mulai = \$this->input->post('mulai');
24	\$akhir = \$this->input->post('akhir');
25	
26	\$d = \$this->LaporanModel->getLaporan(\$mulai, \$akhir)->result();
27	\$jumlahData = \$this->LaporanModel->getLaporan(\$mulai,
28	\$akhir)->num_rows();
29	
30	
31	if (\$jumlahData == 0) {
32	echo 'data tidak ditemukan';
33	} else {
34	\$no = 1;
35	\$tabel = '<div class="table-responsive mb-3">
36	<table class="table table-hover">
37	<thead>
38	<tr>
39	<th>No. </th>
40	<th>Kendaraan</th>
41	<th>Plat</th>
42	<th>Tanggal</th>

43	<code><th>Jam</th></code>
44	<code><th>Income</th></code>
45	<code></tr></code>
46	<code></thead></code>
47	<code><tbody>';</code>
48	
49	<code>foreach (\$d as \$d) {</code>
50	<code> \$label = \$label . '<tr><td>' . \$no++ . '</td><td>' . \$d->kendaraan .</code>
51	<code>'</td><td>' . \$d->plat . '</td><td>' . \$d->tanggal . '</td><td>' . \$d->jam . '</td><td>' .</code>
52	<code>number_format(\$d->uang_masuk, 0, ',', '.') . '</td></tr>';</code>
53	<code> }</code>
54	
55	<code> \$label = \$label . '</tbody></table><div class="float-right mr-3"><a</code>
56	<code>onclick="exportToExcel()" href="javascript:void(0);" class="btn btn-dark"><i</code>
57	<code>class="fas fa-file-excel"></i> Download</div></div>';</code>
58	
59	<code> echo \$label;</code>
60	<code> }</code>
61	<code>}</code>

Berdasarkan Tabel 4.30 ditampilkan implementasi kode program untuk halaman melihat data rekapan keuangan yang berfungsi mengatur tampilan serta fungsionalitas proses menampilkan data rekapan keuangan kepada pengguna. Penjelasan terkait struktur dan fungsi dari setiap bagian kode tersebut dijabarkan pada Tabel 4.30 sebagai berikut:

Tabel 4.31 Penjelasan Kode Tampilan Melihat Data Rekapan Keuangan

Baris Kode	Deskripsi Kode
1-3	Membuat controller bernama Laporan, yang digunakan untuk mengatur tampilan dan data laporan pencucian.
4-11	Konstruktor ini dijalankan otomatis saat class dipanggil. Model LaporanModel dan UserModel dimuat, serta dicek apakah user sudah login. Kalau belum, akan diarahkan ke halaman login.
12-15	Fungsi ini menampilkan halaman awal laporan. Ia hanya menampilkan tampilan kosong (tanpa data) dari halaman laporan.
16	Fungsi ini bertugas mengambil data laporan berdasarkan rentang tanggal yang diminta pengguna (admin).
17-18	Sistem membaca input tanggal awal dan akhir dari form yang

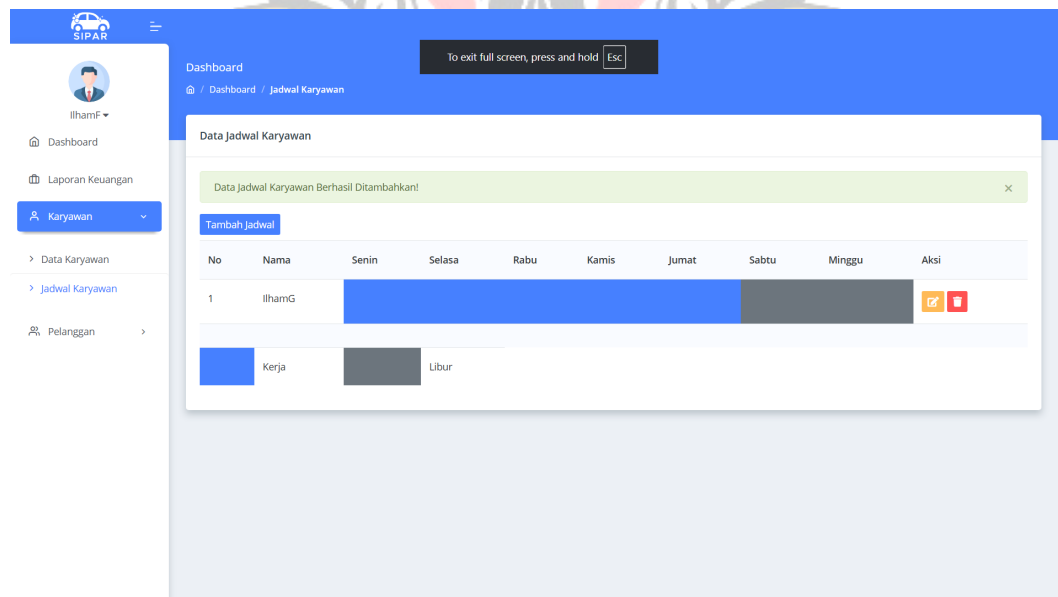
	dikirim oleh admin.
19-20	Mengambil data laporan dari database sesuai tanggal tersebut. Data diambil dari LaporanModel.
22-23	Jika tidak ada data ditemukan dalam tanggal tersebut, akan ditampilkan pesan "data tidak ditemukan".
24-39	Jika data ada, maka sistem akan membuat tabel HTML yang menampilkan semua data pencucian: nomor, kendaraan, plat, tanggal, jam, dan uang masuk.
40	Tabel tersebut dikirim ke browser agar bisa ditampilkan di halaman laporan.

4.5.10 Perancangan Antarmuka Melihat Jadwal Karyawan (SQ_0012)

4.5.1.1 Spesifikasi fungsi Melihat Jadwal Karyawan

Identifikasi nama : Melihat Jadwal Karyawan

Deskripsi inti : Fungsi ini digunakan untuk menampilkan jadwal karyawan



Gambar 4.19 Melihat Jadwal Karyawan

4.5.1.2 Spesifikasi Algoritma

Tabel 4.32 Tabel Algoritma Melihat Jadwal Karyawan

No.	Kode
1	class Jadwal extends CI_Controller
2	{
3	public function __construct()
4	{
5	parent::__construct();
6	
7	\$this->load->model('JadwalModel');
8	\$this->load->model('KaryawanModel');
9	
10	if (\$this->session->userdata('nama') == NULL) {
11	redirect(base_url('login'));
12	}
13	}
14	
15	public function index()
16	{
17	\$data['jadwal'] = \$this->JadwalModel->getJadwal();
18	\$this->load->view('admin/template/header');
19	\$this->load->view('admin/jadwal', \$data);
20	\$this->load->view('admin/template/footer');
21	}
22	
23	public function viewJadwal()
24	{
25	\$data['jadwal'] = \$this->JadwalModel->getJadwal();
26	\$this->load->view('karyawan/template/header');
27	\$this->load->view('karyawan/jadwal', \$data);
28	\$this->load->view('karyawan/template/footer');
29	}

Berdasarkan Tabel 4.31 ditampilkan implementasi kode program untuk halaman *Login* yang berfungsi mengatur tampilan serta fungsionalitas proses autentikasi pengguna. Penjelasan terkait struktur dan fungsi dari setiap bagian kode tersebut dijabarkan pada Tabel 4. sebagai berikut:

Tabel 4.33 Penjelasan Kode Tampilan Melihat Jadwal Karyawan

Baris Kode	Deskripsi Kode
1	Membuat controller baru bernama Jadwal yang merupakan bagian dari CodeIgniter. Fungsinya untuk mengatur jadwal kerja.

2-10	Fungsi ini akan otomatis dijalankan saat class dipanggil. Digunakan untuk: memuat model JadwalModel dan KaryawanModel, serta mengecek apakah user sudah login. Jika belum login, akan diarahkan ke halaman login.
12-16	Fungsi ini digunakan untuk menampilkan halaman daftar jadwal kerja karyawan bagi admin. Data jadwal diambil dari model lalu ditampilkan lewat tiga bagian tampilan (header, konten, footer).
18-22	Fungsi ini menampilkan jadwal kerja versi karyawan. Jadi, tampilannya ditujukan khusus untuk akun karyawan yang sedang login.

4.6 Testing

Pengujian dilaksanakan untuk memverifikasi fungsionalitas sistem SIPAR berbasis website, dengan melakukan pemeriksaan menyeluruh pada setiap komponen yang diuji. Langkah ini bertujuan untuk meminimalisir terjadinya kesalahan (error) dan memastikan bahwa keluaran yang dihasilkan sesuai dengan kebutuhan dan harapan pengguna. Hasil pengujian tersebut didokumentasikan pada Lampiran D (DUPL).

4.6.1 Rencana Pengujian

Strategi rencana pengujian perangkat lunak SIPAR meliputi pengujian validasi, yaitu pengujian terhadap kebutuhan atau *requirement system* yang telah didefinisikan di dalam dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL). Penjelasan lebih rinci mengenai rencana pengujian tersebut disajikan pada tabel berikut.

Tabel 4.34 Rencana Pengujian

Kelas Uji	Butir Uji	SKPL	Tingkat Pengujian	Jenis Pengujian	Jadwal
-----------	-----------	------	-------------------	-----------------	--------

Pengujian menambahkan pesanan	Pengujian basis path menambahkan pesanan	SRS_F-SIPAR-011	Pengujian Unit	Black Box	29/12/2020
Pengujian menambahkan data keuangan	Pengujian basis path menambahkan data keuangan	SRS_F-SIPAR-007	Pengujian Unit	Black Box	29/12/2020
Pengujian pencatatan plat kendaraan dan rekap data kendaraan yang berkunjung	Semua field dikosongkan	SRS_F-SIPAR-07	Pengujian Validasi	Black Box	29/12/2020
	Salah satu field dikosongkan	SRS_F-SIPAR-07	Pengujian Validasi	Black Box	29/12/2020
	Field diisi selain angka	SRS_F-SIPAR-07	Pengujian Validasi	Black Box	29/12/2020
	Field diisi angka atau number	SRS_F-SIPAR-07	Pengujian Validasi	Black Box	29/12/2020
Pengujian reservasi pencucian kendaraan	Semua field dikosongkan	SRS_F-SIPAR-011	Pengujian Validasi	Black Box	29/12/2020

	Salah satu field dikosongkan	SRS_F-SIPAR-01 1	Pengujian Validasi	Black Box	29/12 /2020
	Field diisi selain angka	SRS_F-SIPAR-01 1	Pengujian Validasi	Black Box	29/12 /2020
	Field diisi angka atau number		Pengujian Validasi	Black Box	29/12 /2020
Pengujian registrasi akun untuk pelanggan	Semua field dikosongkan	SRS_F-SIPAR-001	Pengujian Validasi	Black Box	29/12 /2020
	Semua field dikosongkan	SRS_F-SIPAR-001	Pengujian Validasi	Black Box	29/12 /2020
	Field diisi selain angka	SRS_F-SIPAR-001	Pengujian Validasi	Black Box	29/12 /2020

	Field diisi angka atau number	SRS_F-SIPAR-001	Pengujian Validasi	Black Box	29/12/2020
Pengujian Constraint	Melihat Program dengan Bahasa PHP	SRS_NF-SIPAR-006	Pengujian Validasi	Black Box	29/12/2020
Pengujian Peforma	Sistem dapat memberikan layanan kepada pengguna tidak lebih dari 1 menit	SRS_NF-SIPAR-003	Pengujian Validasi	Black Box	29/12/2020

4.6.2 Hasil Uji

Berikut merupakan pengujian yang dilakukan terhadap SIPAR. Pengujian dilakukan menggunakan scenario usecase yang telah dijelaskan.

Tabel 4.35 Tabel Hasil Uji

No	No Req	Test Name	Test Case	Expected Result	Result	Status
----	--------	-----------	-----------	-----------------	--------	--------

1	SRS_F-SIP AR-07	Pengujian pencatatan plat kendaraan dan rekap data kendaraan yang berkunjung	Semua Field Kosong	Sistem akan menampilkan "Please fill out this field" pada semua field yang dibiarkan kosong.	Sistem akan menampilkan "Please fill out this field" pada semua field yang dibiarkan kosong.	Valid
2	SRS_F-SIP AR-07	Pengujian pencatatan plat kendaraan dan rekap data kendaraan yang berkunjung	Salah Satu Field Kosong	Sistem akan menampilkan "Please fill out this field" pada field yang telah dibiarkan kosong.	Sistem akan menampilkan "Please fill out this field" pada field yang telah dibiarkan kosong.	Valid
3	SRS_F-SIP AR-07	Pengujian pencatatan plat kendaraan dan rekap data kendaraan yang berkunjung	Field diisi selain angka	Sistem akan menampilkan "Please fill out this field". Yang berarti pada field jumlah pembayaran harus diisi dengan angka atau number	Sistem akan menampilkan "Please fill out this field". Yang berarti pada field jumlah pembayaran harus diisi dengan angka atau number	Valid
4	SRS_F-SIP AR-07	Pengujian pencatatan plat kendaraan dan rekap data kendaraan yang berkunjung	Field diisi Angka atau Number	Sistem akan menampilkan bahwa data yang telah diinput berhasil disimpan. Bukti bahwa data	Sistem akan menampilkan bahwa data yang telah diinput berhasil disimpan. Bukti bahwa data	Valid

				telah tersimpan dapat dilihat pada menu laporan keuangan.	telah tersimpan dapat dilihat pada menu laporan keuangan.	
5	SRS_F-SIP AR-011	Pengujian reservasi pencucian kendaraan	Semua Field Kosong	Sistem akan menampilkan "Please fill out this field" pada semua field yang dibiarkan kosong.	Sistem akan menampilkan "Please fill out this field" pada semua field yang dibiarkan kosong.	Valid
6	SRS_F-SIP AR-011	Pengujian reservasi pencucian kendaraan	Salah Satu Field Kosong	Sistem akan menampilkan "Please fill out this field" pada salah satu field yang dibiarkan kosong.	Sistem akan menampilkan "Please fill out this field" pada salah satu field yang dibiarkan kosong.	Valid
7	SRS_F-SIP AR-011	Pengujian reservasi pencucian kendaraan	Field diisi selain angka	Sistem akan menampilkan "Jadwal Reservasi anda berhasil di Booking!" yang berarti reservasi telah terdaftar di admin.	Sistem akan menampilkan "Jadwal Reservasi anda berhasil di Booking!" yang berarti reservasi telah terdaftar di admin.	Valid
8	SRS_F-SIP AR-011	Pengujian reservasi pencucian kendaraan	Field diisi Angka atau Number	Sistem akan menampilkan "Jadwal Reservasi anda berhasil di Booking!" yang berarti reservasi telah terdaftar di admin.	Sistem akan menampilkan "Jadwal Reservasi anda berhasil di Booking!" yang berarti reservasi telah terdaftar di admin.	Valid
9	SRS_F-SI PAR-001	Pengujian registrasi akun untuk pelanggan	Semua Field Kosong	Sistem akan menampilkan "Please fill out this field" pada semua field yang dibiarkan kosong.	Sistem akan menampilkan "Please fill out this field" pada semua field yang dibiarkan kosong.	Valid
10	SRS_F-SI PAR-001	Pengujian registrasi akun	Salah Satu Field Kosong	Sistem akan menampilkan "Please fill out this	Sistem akan menampilkan "Please fill out this	Valid

		untuk pelanggan		field” pada field yang dibiarkan kosong.	field” pada field yang dibiarkan kosong.	
11	SRS_F-SI PAR-001	Pengujian registrasi akun untuk pelanggan	Field diisi selain angka	Sistem menampilkan “Please include an ‘@’ in email address. ‘SIPAR’ is missing ‘@’.” Pada field email, yang field tersebut harus menyertakan ‘@’ pada inputan.	Sistem menampilkan “Please include an ‘@’ in email address. ‘SIPAR’ is missing ‘@’.” Pada field email, yang field tersebut harus menyertakan ‘@’ pada inputan.	Valid
12	SRS_F-SI PAR-001	Pengujian registrasi akun untuk pelanggan	Field diisi Angka atau Number	Sistem menampilkan “Please include an ‘@’ in email address. ‘SIPAR’ is missing ‘@’.” Pada field email, yang field tersebut harus menyertakan ‘@’ pada inputan.	Sistem menampilkan “Please include an ‘@’ in email address. ‘SIPAR’ is missing ‘@’.” Pada field email, yang field tersebut harus menyertakan ‘@’ pada inputan.	Valid
13	SRS_NF-SI PAR-006	Pengujian Constraint	Melihat Program dengan bahasa PHP	Ditemukan file pada sistem SIPAR yang menggunakan .php dan mengandung bahasa pemrograman PHP	Ditemukan file pada sistem SIPAR yang menggunakan .php dan mengandung bahasa pemrograman PHP	Valid
14	SRS_NF-SI PAR-003	Pengujian Performa	Sistem memberikan layanan kepada pengguna tidak boleh lebih dari 1 detik	Sistem dapat mengakses tampilan per halaman kurang atau sama dengan 1 detik	Sistem dapat mengakses tampilan per halaman kurang atau sama dengan 1 detik	Valid

4.7 Deployment

Setelah dilakukan pengujian yang memvalidasi bahwa sistem siap digunakan, selanjutnya akan dilanjutkan pada tahap deployment. Pada tahap ini sistem akan dilanjutkan untuk dilakukan hosting. Proses hosting dilakukan untuk mempublikasikan aplikasi sistem informasi ke internet agar dapat diakses oleh pengguna secara luas. Dalam penelitian ini, layanan hosting dari Rumahweb digunakan karena menyediakan fitur yang mendukung kebutuhan aplikasi berbasis PHP dan database MySQL. Tahapan hosting dimulai dengan melakukan pendaftaran domain dan layanan hosting pada situs resmi Rumahweb.

