BAB 4
IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi Sistem

Implementasi sistem merupakan suatu tahap penerapan perancangan berdasarkan hasil analisis ke dalam bentuk kode-kode pemograman tertentu untuk menghasilkan perangkat lunak yang dibangun pada lingkungan yang sesungguhnya. Adapun pembahasan implementasi sistem terdiri dari spesifikasi perangkat keras, spesifikasi perangkat lunak, implementasi basis data, implementasi class, dan implementasi antarmuka.

4.1.1 Perangkat Keras Pembangun

Dalam penelitian ini, dibutuhkan perangkat keras untuk menerapkan suatu sistem yang dapat mengimplementasikan RFID sebagai akses ke aplikasi. Adapun spesifikasi perangkat keras yang digunakan adalah seperti pada Tabel 4.1 dan pada Tabel 4.2.

Tabel 4.1 Spesifikasi Perangkat Keras Server

Perangkat Keras	**Spesifikasi**
Processor	Intel® Core i3 2100, 3.10 GHz
Display	1366 x 768 *pixels*
Memory	4 GB
Storage	500 GB
RFID Reader	125 Khz

Tabel 4.2 Spesifikasi Perangkat Keras Client

Perangkat Keras	**Spesifikasi**
Processor	Intel® Core i3 2100, 3.10 GHz
Display	1366 x 768 *pixels*
Memory	2 GB
Storage	256 GB
RFID Reader	125 z

4.1.2 Perangkat Lunak Pembangun

Selain perangkat keras, pada penelitian ini dibutuhkan juga perangkat lunak untuk mendukung dalam pengimplementasian QR *code* sebagai tiket masuk. Adapun spesifikasi perangkat lunak yang digunakan adalah seperti pada Tabel 4.3.

Tabel 4.3 Spesifikasi Perangkat Lunak

Perangkat Lunak	**Spesifikasi**
Sistem Operasi PC	Microsoft Windows 8.1 Pro x86
Sistem Operasi Android	Android versi 4.4.4 KitKat

DBMS	MySql v5.5.24
Tools Pembangun	Visual Studio 2015 Notepad++ WampServer v2.4

4.1.3 Implementasi Basis Data

Dalam penelitian ini, implementasi basis data dilakukan dengan menggunakan DBMS MySql. Implementasi basis data dalam bahasa SQL adalah seperti pada Tabel 4.4.

Tabel 4.4 Implementasi Basis Data pada Bahasa SQL

Nama Tabel	Perintah
absensis	CREATE TABLE IF NOT EXISTS `absensis` (`id` int(10) unsigned NOT NULL AUTO_INCREMENT, `rfid` varchar(255) COLLATE utf8_unicode_ci NOT NULL, `jam_masuk` timestamp NULL DEFAULT NULL, `jam_keluar` timestamp NULL DEFAULT NULL, `created_at` timestamp NULL DEFAULT NULL, `updated_at` timestamp NULL DEFAULT NULL, PRIMARY KEY (`id`), KEY `absensis_rfid_foreign` (`rfid`)) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci AUTO_INCREMENT=1 ;
buku	CREATE TABLE IF NOT EXISTS `buku_perpustakaans` (`id` int(10) unsigned NOT NULL AUTO_INCREMENT, `isbn` varchar(255) COLLATE utf8_unicode_ci DEFAULT NULL, `judul_buku` varchar(255) COLLATE utf8_unicode_ci NOT NULL, `seri` varchar(255) COLLATE utf8_unicode_ci DEFAULT NULL, `edisi` varchar(255) COLLATE utf8_unicode_ci DEFAULT NULL, `tahun` varchar(255) COLLATE utf8_unicode_ci DEFAULT NULL, `jumlah` int(11) DEFAULT NULL, `fullfillment` int(11) DEFAULT NULL, `satuan` varchar(255) COLLATE utf8_unicode_ci DEFAULT NULL, `created_at` timestamp NULL DEFAULT NULL, `updated_at` timestamp NULL DEFAULT NULL, PRIMARY KEY (`id`)) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci AUTO_INCREMENT=1 ;
detail_peminjaman	CREATE TABLE IF NOT EXISTS `detail_peminjamen` (`id` int(10) unsigned NOT NULL AUTO_INCREMENT,

	`id_transaksi` int(10) unsigned NOT NULL, `id_buku` int(10) unsigned NOT NULL, `tanggal_pinjam` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP, `tanggal_kembali` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00', `lama_pinjam` int(11) NOT NULL, `satuan_lama` varchar(255) COLLATE utf8_unicode_ci NOT NULL, `jumlah` int(11) NOT NULL, `satuan` varchar(255) COLLATE utf8_unicode_ci NOT NULL, `harga` bigint(20) NOT NULL, `created_at` timestamp NULL DEFAULT NULL, `updated_at` timestamp NULL DEFAULT NULL, PRIMARY KEY (`id`), KEY `detail_peminjamen_id_buku_foreign` (`id_buku`), KEY `detail_peminjamen_id_transaksi_foreign` (`id_transaksi`)) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci AUTO_INCREMENT=1 ;
detail_transaksis	CREATE TABLE IF NOT EXISTS `detail_transaksis` (`id` int(10) unsigned NOT NULL AUTO_INCREMENT, `id_transaksi` int(10) unsigned NOT NULL, `jumlah` int(11) NOT NULL, `satuan` varchar(255) COLLATE utf8_unicode_ci NOT NULL, `harga` bigint(20) NOT NULL, `created_at` timestamp NULL DEFAULT NULL, `updated_at` timestamp NULL DEFAULT NULL, PRIMARY KEY (`id`), KEY `detail_transaksis_id_transaksi_foreign` (`id_transaksi`)) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci AUTO_INCREMENT=1 ;
gurus	CREATE TABLE IF NOT EXISTS `gurus` (`nip` varchar(255) COLLATE utf8_unicode_ci NOT NULL, `nama` varchar(50) COLLATE utf8_unicode_ci NOT NULL, `tempat_lahir` varchar(100) COLLATE utf8_unicode_ci NOT NULL, `tanggal_lahir` date DEFAULT NULL, `alamat` varchar(250) COLLATE utf8_unicode_ci DEFAULT NULL, `telepon` varchar(16) COLLATE utf8_unicode_ci DEFAULT NULL, `spesialisasi` varchar(50) COLLATE

	utf8_unicode_ci DEFAULT NULL, `created_at` timestamp NULL DEFAULT NULL, `updated_at` timestamp NULL DEFAULT NULL, PRIMARY KEY (`nip`)) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
kelas	CREATE TABLE IF NOT EXISTS `kelas` (`id` int(10) unsigned NOT NULL AUTO_INCREMENT, `nama_kelas` varchar(10) COLLATE utf8_unicode_ci NOT NULL, `wali_kelas` varchar(255) COLLATE utf8_unicode_ci DEFAULT NULL, `keterangan` varchar(100) COLLATE utf8_unicode_ci DEFAULT NULL, `created_at` timestamp NULL DEFAULT NULL, `updated_at` timestamp NULL DEFAULT NULL, PRIMARY KEY (`id`), KEY `kelas_wali_kelas_foreign` (`wali_kelas`)) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci AUTO_INCREMENT=1 ;
kelas_siswas	CREATE TABLE IF NOT EXISTS `kelas_siswas` (`id_kelas` int(10) unsigned NOT NULL, `nis` varchar(9) COLLATE utf8_unicode_ci NOT NULL, `isActive` tinyint(1) NOT NULL DEFAULT '1', `created_at` timestamp NULL DEFAULT NULL, `updated_at` timestamp NULL DEFAULT NULL, KEY `kelas_siswas_nis_foreign` (`nis`), KEY `kelas_siswas_id_kelas_foreign` (`id_kelas`)) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
ketidakhadirans	CREATE TABLE IF NOT EXISTS `ketidakhadirans` (`id` int(10) unsigned NOT NULL AUTO_INCREMENT, `nis` varchar(9) COLLATE utf8_unicode_ci NOT NULL, `tahun` int(11) NOT NULL, `bulan` int(11) NOT NULL, `keterangan` varchar(255) COLLATE utf8_unicode_ci DEFAULT NULL, `created_at` timestamp NULL DEFAULT NULL, `updated_at` timestamp NULL DEFAULT NULL, PRIMARY KEY (`id`), KEY `ketidakhadirans_nis_foreign` (`nis`)) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci AUTO_INCREMENT=1 ;
parkirs	CREATE TABLE IF NOT EXISTS `parkirs` (`id` int(10) unsigned NOT NULL AUTO_INCREMENT, `rfid` varchar(255) COLLATE utf8_unicode_ci NOT NULL,

	`jam_masuk` timestamp NULL DEFAULT NULL, `jam_keluar` timestamp NULL DEFAULT NULL, `biaya` bigint(20) DEFAULT NULL, `nopol` varchar(9) COLLATE utf8_unicode_ci DEFAULT NULL, `created_at` timestamp NULL DEFAULT NULL, `updated_at` timestamp NULL DEFAULT NULL, PRIMARY KEY (`id`), KEY `parkirs_rfid_foreign` (`rfid`)) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci AUTO_INCREMENT=1 ;
password_resets	CREATE TABLE IF NOT EXISTS `password_resets` (`email` varchar(255) COLLATE utf8_unicode_ci NOT NULL, `token` varchar(255) COLLATEutf8_unicode_ci NOT NULL, `created_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP, KEY `password_resets_email_index` (`email`), KEY `password_resets_token_index` (`token`)) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
pengunjung_pepuses	CREATE TABLE IF NOT EXISTS `pengunjung_perpuses` (`id` int(10) unsigned NOT NULL AUTO_INCREMENT, `rfid` varchar(255) COLLATE utf8_unicode_ci NOT NULL, `jam-masuk` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP, `jam-keluar` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00', `created_at` timestamp NULL DEFAULT NULL, `updated_at` timestamp NULL DEFAULT NULL, PRIMARY KEY (`id`), KEY `pengunjung_perpuses_rfid_foreign` (`rfid`)) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci AUTO_INCREMENT=1 ;
rekap_absensis	CREATE TABLE IF NOT EXISTS `rekap_absensis` (`id` int(10) unsigned NOT NULL AUTO_INCREMENT, `nis` varchar(9) COLLATE utf8_unicode_ci NOT NULL, `tahun` int(11) NOT NULL, `bulan` int(11) NOT NULL, `sakit` int(11) NOT NULL, `izin` int(11) NOT NULL, `tanpa_keterangan` int(11) NOT NULL, `jumlah_masuk` int(11) NOT NULL, `created_at` timestamp NULL DEFAULT NULL, `updated_at` timestamp NULL DEFAULT NULL,

	PRIMARY KEY (`id`), KEY `rekap_absensis_nis_foreign` (`nis`)) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci AUTO_INCREMENT=1 ;
roles	CREATE TABLE IF NOT EXISTS `roles` (`id` int(10) unsigned NOT NULL AUTO_INCREMENT, `name` varchar(255) COLLATE utf8_unicode_ci NOT NULL, `created_at` timestamp NULL DEFAULT NULL, `updated_at` timestamp NULL DEFAULT NULL, PRIMARY KEY (`id`)) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci AUTO_INCREMENT=1 ;
siswas	CREATE TABLE IF NOT EXISTS `siswas` (`nis` varchar(9) COLLATE utf8_unicode_ci NOT NULL, `nama` varchar(50) COLLATE utf8_unicode_ci NOT NULL, `tempat_lahir` varchar(50) COLLATE utf8_unicode_ci NOT NULL, `tanggal_lahir` date DEFAULT NULL, `alamat` varchar(250) COLLATE utf8_unicode_ci NOT NULL, `tahun_masuk` int(11) NOT NULL, `created_at` timestamp NULL DEFAULT NULL, `updated_at` timestamp NULL DEFAULT NULL, PRIMARY KEY (`nis`)) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
smart_cards	CREATE TABLE IF NOT EXISTS `smart_cards` (`rfid` varchar(255) COLLATE utf8_unicode_ci NOT NULL, `nis` varchar(9) COLLATE utf8_unicode_ci NOT NULL, `nopol` varchar(11) COLLATE utf8_unicode_ci DEFAULT NULL, `punya_sim` tinyint(1) NOT NULL DEFAULT '0', `saldo` bigint(20) DEFAULT NULL, `status` varchar(255) COLLATE utf8_unicode_ci NOT NULL DEFAULT 'active', `created_at` timestamp NULL DEFAULT NULL, `updated_at` timestamp NULL DEFAULT NULL, PRIMARY KEY (`rfid`), KEY `smart_cards_nis_foreign` (`nis`)) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
transaksis	CREATE TABLE IF NOT EXISTS `transaksis` (`id` int(10) unsigned NOT NULL AUTO_INCREMENT, `rfid` varchar(255) COLLATE utf8_unicode_ci NOT NULL, `tanggal_transaksi` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,

	`jenis_transaksi` varchar(255) COLLATE utf8_unicode_ci NOT NULL, `debit_credit` varchar(255) COLLATE utf8_unicode_ci NOT NULL, `nominal` bigint(20) NOT NULL, `jumlah_item` int(11) NOT NULL, `created_at` timestamp NULL DEFAULT NULL, `updated_at` timestamp NULL DEFAULT NULL, PRIMARY KEY (`id`), KEY `transaksis_rfid_foreign` (`rfid`)) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci AUTO_INCREMENT=1 ;
users	CREATE TABLE IF NOT EXISTS `users` (`id` int(10) unsigned NOT NULL AUTO_INCREMENT, `name` varchar(255) COLLATE utf8_unicode_ci NOT NULL, `email` varchar(255) COLLATE utf8_unicode_ci NOT NULL, `password` varchar(60) COLLATE utf8_unicode_ci NOT NULL, `remember_token` varchar(100) COLLATE utf8_unicode_ci DEFAULT NULL, `created_at` timestamp NULL DEFAULT NULL, `updated_at` timestamp NULL DEFAULT NULL, PRIMARY KEY (`id`), UNIQUE KEY `users_email_unique` (`email`)) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci AUTO_INCREMENT=2 ;
user_roles	CREATE TABLE IF NOT EXISTS `user_roles` (`user_id` int(10) unsigned NOT NULL, `role_id` int(10) unsigned NOT NULL, `created_at` timestamp NULL DEFAULT NULL, `updated_at` timestamp NULL DEFAULT NULL, KEY `user_roles_user_id_foreign` (`user_id`), KEY `user_roles_role_id_foreign` (`role_id`)) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

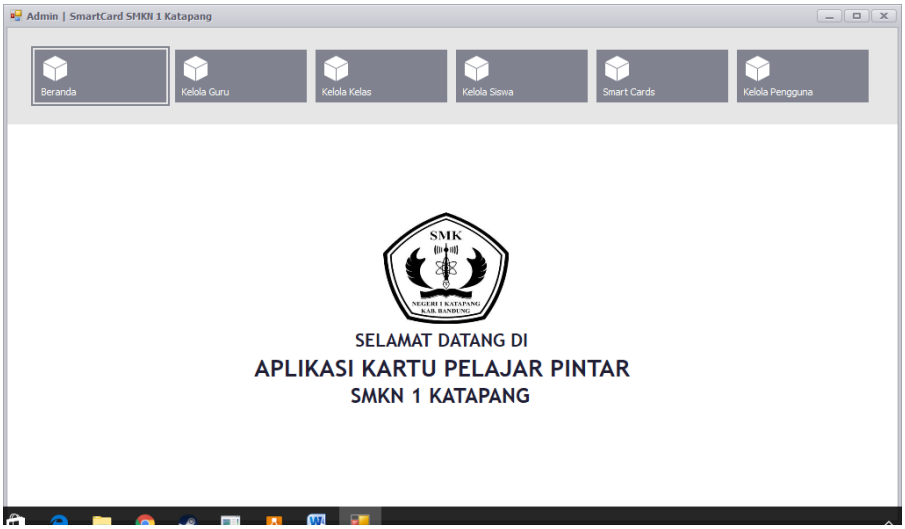
4.1.4 Implementasi Antarmuka

Dalam penelitian ini sistem terdiri dari 4 aplikasi yaitu aplikasi kartu pelajar, aplikasi parkir, aplikasi perpustakaan dan aplikasi absensi. Seperti sudah dijelaskan sebelumnya aplikasi aplikasi ini berbasis desktop.

4.1.4.1 Antarmuka Kartu Pelajar

Antarmuka dari aplikasi kartu pelajar diimplementasikan dalam bentuk desktop, dengan membuka aplikasi kartu pelajar dan melakukan login, maka akan langsung terbuka beranda dari aplikasi kartu pelajar. Contoh antarmuka karu

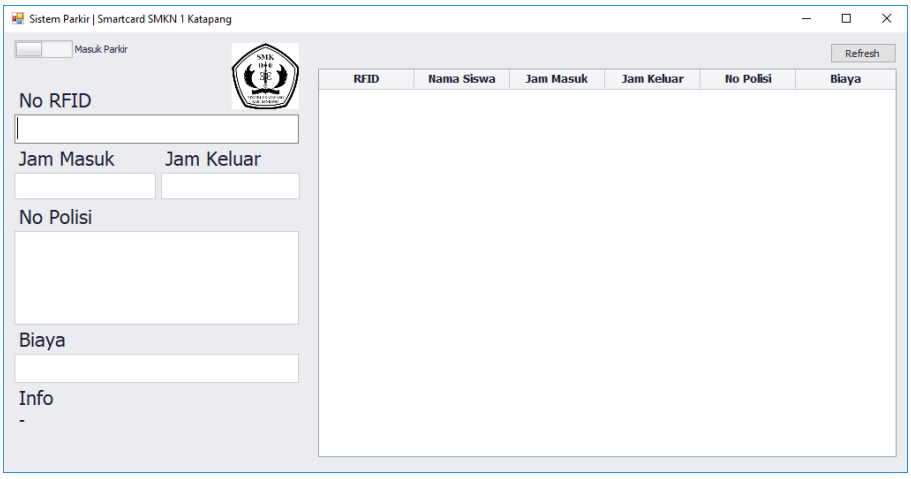
pelajar adalah seperti gambar. Untuk implementasi lebih lengkapnya dapat dilihat pada lampiran.



Gambar 4. 1 Halaman Utama Aplikasi Kartu Pelajar

4.1.4.2 Antarmuka Parkir

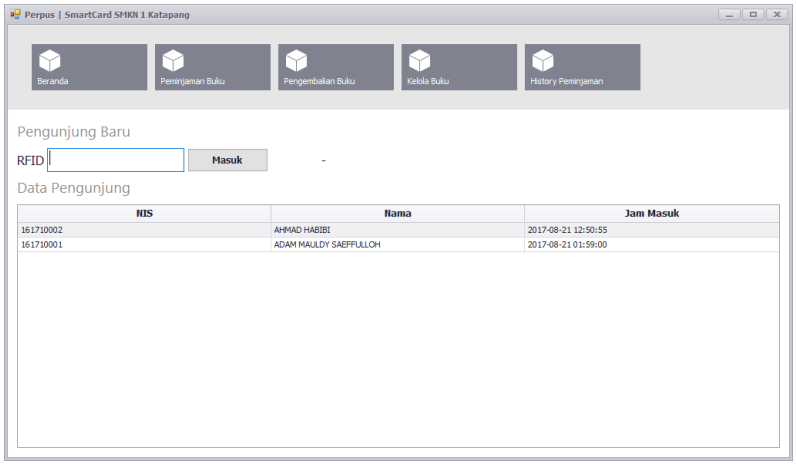
Antarmuka dari aplikasi kartu pelajar diimplementasikan dalam bentuk desktop, dengan membuka aplikasi kartu pelajar dan melakukan login, maka akan langsung terbuka beranda dari aplikasi kartu pelajar. Contoh antarmuka karu pelajar adalah seperti gambar. Untuk implementasi lebih lengkapnya dapat dilihat pada lampiran.



Gambar 4. 2 Antarmuka Parkir

4.1.4.3 Antarmuka Perpustakaan

Antarmuka dari aplikasi kartu pelajar diimplementasikan dalam bentuk desktop, dengan membuka aplikasi kartu pelajar dan melakukan login, maka akan langsung terbuka beranda dari aplikasi kartu pelajar. Contoh antarmuka karu pelajar adalah seperti gambar. Untuk implementasi lebih lengkapnya dapat dilihat pada lampiran.



Gambar 4. 3 Antarmuka Perpustakaan

4.1.4.4 Antarmuka Absensi

Antarmuka dari aplikasi kartu pelajar diimplementasikan dalam bentuk desktop, dengan membuka aplikasi kartu pelajar dan melakukan login, maka akan langsung terbuka beranda dari aplikasi kartu pelajar. Contoh antarmuka karu pelajar adalah seperti gambar. Untuk implementasi lebih lengkapnya dapat dilihat pada lampiran.



Gambar 4. 4 Antarmuka Absensi

4.1.5 Implementasi Class

Implementasi *class* merupakan implementasi dari perancangan diagram kelas yang terbentuk berdasarkan diagram *use case* kedalam bentuk fisik file dalam pembuatan dan jalannya proses dalam sistem. Dalam penelitian ini digunakan dua jenis bahasa pemograman yaitu php dan java sehingga ekstensi dari file tersebut ada yang berekstensi .php dan .java. Untuk lebih lengkapnya, dapat dilihat pada Tabel 4.5.

Tabel 4.5 Implementasi Class

No.	Nama Kelas	Nama File
1	Absensi	Absensi.php
2	Buku	Buku.php
3	DetailPeminjaman	DetailPeminjaman.php
4	DetailTransaksi	DetailTransaksi.php
5	Parkir	Parkir.php
6	RekapAbsensi	RekapAbsensi.php
7	SmartCard	SmartCard.php
8	Transaksi	Transaksi.php
9	Kelas	Kelas.php
10	KelasSiswa	KelasSiswa.php
11	Guru	Guru.php
12	Siswa	Siswa.php
13	UserRole	UserRole.php
14	PengunjungPerpus	PengunjungPerpus.php
15	Ketidakhadiran	Ketidakhadiran.php
16	User	User.php
17	Role	Role.php
18	JumlahBuku	JumlahBuku.php
19	NomorPolisi	NomorPolisi.php

20	SiswaController	SiswaController.php
21	GuruController	GuruController.php
22	KelasController	KelasController.php
23	KelasSiswaController	KelasSiswaController.php
24	AbsensiCOntroller	AbsensiCOntroller.php
25	BukuController	BukuController.php
26	SmartCardController	SmartCardController.php
27	ParkirController	ParkirController.php
28	DetailController	DetailController.php
29	DetailPeminjamanController	DetailPeminjamanController.php
30	PinjamBukuController	PinjamBukuController.php
31	PengunjungPerpusController	PengunjungPerpusController.php
32	FormKelolaSiswa	FormKelolaSiswa.cs
33	FormKelolaKelas	FormKelolaKelas.cs
34	FormKelolaKelasSiswa	FormKelolaKelasSiswa.cs
35	FormTopUpSaldo	FormTopUpSaldo.cs
36	FormKelolaKartuPintar	FormKelolaKartuPintar.cs
37	FormAbsensi	FormAbsensi.cs
38	FormRekapitulasiAbsensi	FormRekapitulasiAbsensi.cs
39	FormMasukParkir	FormMasukParkir.cs
40	FormKeluarParkir	FormKeluarParkir.cs
41	FormListParkir	FormListParkir.cs
42	FormDataPengunjung	FormDataPengunjung.cs
43	FormPeminjamanBuku	FormPeminjamanBuku.cs
44	FormPengembalianBuku	FormPengembalianBuku.cs
45	FormKelolaBuku	FormKelolaBuku.cs
46	FormLogin	FormLogin.cs
47	FormKelolaGuru	FormKelolaGuru.cs
48	FormDaftarSIM	FormDaftarSIM.cs

4.2 Pengujian Sistem

Pengujian sistem merupakan hal yang perlu dilakukan dalam penelitian ini dengan tujuan untuk menemukan kesalahan atau kekurangan yang ada pada sistem yang diuji. Pengujian sistem dimaksudkan untuk mengetahui apakah sistem yang dibuat sudah memenuhi kriteria, sesuai dengan tujuan akan dicapai.

4.2.1 Pengujian Alpha

Pengujian *alpha* merupakan pengujian terhadap fungsional pada sistem. Pengujian *alpha* ini dilakukan untuk menguji suatu sistem baru. Pengujian *alpha* sendiri berfokus pada persyaratan fungsional dari sistem.

4.2.1.1 Rencana Pengujian Alpha

Pengujian dilakukan dengan menguji setiap fungsional sistem berdasarkan *use case* diagram dan juga kemungkinan kesalahan yang terjadi untuk setiap fungsional. Pengujian ini dilakukan dengan metode pengujian *Black Box* dimana pengujian memperhatikan masukan ke sistem dan keluaran dari sistem. Rincian

dari rencana pengujian yang akan dilakukan pada penelitian ini dapat dilihat pada Tabel 4.6.

Tabel 4.6 Rencana Pengujian Sistem

No	Fungsional yang Diuji	Detail Pengujian	Jenis Uji
1	Login	Memasukan username dan password	*Black Box*
		Memverifikasi data username dan password	
		Menyimpan data *event* baru kedalam *database*	
2	Pembuatan Kartu Pelajar RFID	Membaca ID RFID	*Black Box*
		Menyimpan data kartu pelajar ke dalam *database* meliputi ID RFID, NIS, kepemilikan SIM, Nomor Polisi, dan saldo awal	
		Menampilkan halaman detail kepemilikan kartu pelajar	
3	Kelola data guru	Melakukan update terhadap data guru kedalam *database*	*Black Box*
		Menghapus data guru dari dalam *database*	
		Menampilkan data guru dari *database*	
		Menyimpan data guru baru kedalam *database*	
4	Kelola data siswa	Melakukan update terhadap data siswa kedalam *database*	*Black Box*
		Menghapus data siswa dari dalam *database*	
		Menampilkan data siswa dari *database*	
		Menyimpan data siswa baru kedalam *database*	
5	Kelola data kelas	Melakukan update terhadap data kelas kedalam *database*	*Black Box*
		Menghapus data kelas dari dalam *database*	
		Menampilkan data kelas dari *database*	
		Menyimpan data kelas baru kedalam *database*	
6	Kelola data pengguna	Melakukan update terhadap data pengguna kedalam *database*	*Black Box*
		Menghapus data pengguna dari dalam *database*	
		Menampilkan data pengguna dari *database*	
		Menyimpan data pengguna baru kedalam *database*	

7	Absensi datang	Melakukan validasi ID RFID dari *database*	*Black Box*
		Mencatat waktu absen datang kedalam *database*	
		Melakukan update kehadiran kedalam *database*	
8	Absensi pulang	Melakukan validasi ID RFID dari *database*	*Black Box*
		Mencatat waktu absen pulang kedalam *database*	
		Melakukan update kehadiran kedalam *database*	
9	Keterangan tidak hadir	Menampilkan data absen dari *database*	*Black Box*
		Melakukan update data absen kedalam *database*	
		Menyimpan data absen kedalam *database*	
		Menghapus data absen dari *database*	
10	Peminjaman buku	Memasukan kode buku dan lama pinjam	*Black Box*
		Melakukan validasi ID RFID dari *database*	
		Melakukan update data pinjam buku kedalam *database*	
11	Pengembalian buku	Melakukan validasi ID RFID dari *database*	*Black Box*
		Menampilkan data pinjam dari *database*	
		Melakukan validasi keterlambatan waktu pinjam	
		Mengurangi jumlah saldo jika terlambat dari *database*	
12	Pengunjung perpustakaan	Melakukan validasi ID RFID dari *database*	*Black Box*
		Mencatat waktu mengunjungi perpustakaan kedalam *database*	
13	Parkir masuk	Melakukan validasi ID RFID dari *database*	*Black Box*
		Mencatat waktu parkir masuk ke dalam *database*	
14	Parkir keluar	Melakukan validasi ID RFID dari *database*	*Black Box*
		Menampilkan waktu parkir masuk dari *database*	
		Mencatat waktu parkir keluar kedalam *database*	

4.2.1.2 Kasus dan Hasil Pengujian Alpha

Berdasarkan rencana pengujian, maka dapat dilakukan pengujian *alpha* terhadap sistem ini. Berikut adalah rincian dari kasus dan hasil pengujian *alpha.*

Aplikasi Admin

1) Pembuatan Kartu Pelajar

Pembuatan Kartu Pelajar merupakan fungsional dari sistem yang melayani Pembuatan *smart card* yang dilakukan oleh admin. Tabel 4.7 berikut merupakan hasil pengujian fungsional pembuatan kartu pelajar.

Tabel 4.7 Hasil Pengujian Pembuatan Kartu Pelajar

Kasus dan Hasil Uji (Data Benar)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Masuk ke halaman admin	Menampilkan halaman admin	Sistem menampilkan halaman admin	[√] Diterima [ ] Ditolak
Memilih menu kartu pelajar	Menampilkan halaman kartu pelajar	Sistem menampilkan halaman kartu pelajar	[√] Diterima [ ] Ditolak
Memilih menu tambah	Menampilkan form pembuatan kartu pelajar	Sistem menampilkan form pembuatan kartu pelajar	[√] Diterima [ ] Ditolak
Masukan data pemesanan, contoh masukan : ID RFID : “98732492” NIS : “10110317” SIM : “punya” Nomor Polisi : “D 2739 BP” Saldo Awal :”5000”	Menyimpan data pemesanan kedalam *database* meliputi ID RFID, NIS, SIM, Nomor Polisi, dan Saldo Awal	Form pemesanan terisi dengan data	[√] Diterima [ ] Ditolak
Menekan tombol “*simpan“* dengan data masukan terisi	Menampilkan halaman detail pemesanan dengan data yang sesuai pada saat pembuatan	Sistem menampilkan halaman kartu pelajar dengan data kartu pelajar sesuai dengan yang disubmit	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Memasukan data NIS tidak valid	Tidak menyimpan data pemesanan kedalam *database*	Sistem menampilkan pesan kesalahan “*NIS tidak terdaftar*”	[√] Diterima [ ] Ditolak
Menekan tombol “*simpan”* tanpa	Tidak menyimpan	Sistem	[√] Diterima

data masukan	data pemesanan kedalam *database*	menampilkan pesan kesalahan "*Data belum lengkap*"	[ ] Ditolak
Menekan tombol "*Submit Order"* tanpa data masukan	Tidak menyimpan data tiket yang dipesan kedalam *database*	Sistem menampilkan pesan kesalahan "*Data belum lengkap*"	[√] Diterima [ ] Ditolak

2) Login

Login merupakan fungsional dari sistem untuk melakukan autentifikasi terhadap administrator. Tabel 4.8 berikut merupakan hasil pengujian fungsional login.

Tabel 4.8 Hasil Pengujian Login

Kasus dan Hasil Uji (Data Benar)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Mengakses halaman login admin	Menampilkan halaman login	Sistem menampilkan halaman login admin	[√] Diterima [ ] Ditolak
Menekan tombol login dengan data masukan: *username*: "admin1", *password*: "123adm"	Melakukan autentifikasi administrator, kemudian menampilkan halaman admin dan menjalankan session	Data masukan diterima dan sistem pun menampilkan halaman admin. Session juga dapat dijalankan.	[√] Diterima [ ] Ditolak
Menekan tombol logout	Menghapus session login	Menampilkan kembali halaman login dan session pun terhapus	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Menekan tombol login tanpa data masukan	Tidak dapat login	Sistem tidak login dan menampilkan pesan *"Username atau password belum diisi"*	[√] Diterima [ ] Ditolak
Menekan tombol login dengan data masukan: *username*: "dsfhsyha", *password*: "yt43erg"	Tidak dapat login	Sistem tidak login dan menampilkan pesan *"Login Gagal"*	[√] Diterima [ ] Ditolak

3) Kelola Data Siswa

Kelola data siswa merupakan fungsional dari sistem untuk melakukan pengelolaan data siswa. Tabel 4.9 Berikut merupakan hasil pengujian fungsional kelola data siswa.

Tabel 4.9 Hasil Pengujian Kelola Data Siswa

Kasus dan Hasil Uji (Data Benar)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Mengakses halaman admin	Menampilkan data siswa dari *database*	Menampilkan data daftar siswa dari *database*	[√] Diterima [ ] Ditolak
Menekan tombol "*Create*" dengan data masukan : Nama : "Azis", NIS : "10110288", Tempat Lahir : "Sumedang", Tanggal Lahir: "3 Desember 2002" Alamat : "komplek taman kopo indah I blok 2A" Tahun Masuk : "2016"	Menyimpan data siswa baru kedalam *database*	Data tersimpan kedalam *database* dan muncul pesan konfirmasi "*Data berhasil disimpan*"	[√] Diterima [ ] Ditolak
Menekan tombol "*Update*" dengan mengubah salah satu data siswa	Melakukan update terhadap data siswa kedalam *database*	Data siswa pada *database* berhasil diperbaharui dan muncul pesan konfirmasi "*Update data berhasil*"	[√] Diterima [ ] Ditolak
Menekan tombol "Hapus"	Menghapus data siswa dari dalam *database*	Data siswa terhapus dari *database* dan muncul pesan konfirmasi "*Data berhasil dihapus*"	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Menekan tombol "*Create*" tanpa data masukan	Muncul pesan kesalahan "*Mohon maaf, Semua Input Form Harus Diisi !*"	Sistem menampilkan pesan kesalahan "*Mohon maaf, Semua Input Form Harus Diisi !*"	[√] Diterima [ ] Ditolak
Menekan tombol "*Update*" dengan mengosongkan salah ssatu *field* masukan	Muncul pesan kesalahan "*Mohon maaf, Semua Input Form Harus Diisi !*"	Sistem menampilkan pesan kesalahan "*Mohon maaf, Semua Input Form Harus Diisi !*"	[√] Diterima [ ] Ditolak

4) Kelola Data Pengguna

Kelola data pengguna merupakan fungsional dari sistem untuk melakukan pengelolaan data pengguna. Tabel 4.10 Berikut merupakan hasil pengujian fungsional kelola data pengguna.

Tabel 4.10 Hasil Pengujian Kelola Data Pengguna

Kasus dan Hasil Uji (Data Benar)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Mengakses halaman admin	Menampilkan data pengguna dari *database*	Menampilkan data daftar pengguna dari *database*	[√] Diterima [ ] Ditolak
Menekan tombol "*Create*" dengan data masukan : Nama: Edwin Fajar", Username : "edw8", Password: "*******"	Menyimpan data pengguna baru kedalam *database*	Data tersimpan kedalam *database* dan muncul pesan konfirmasi "*Data berhasil disimpan*"	[√] Diterima [ ] Ditolak
Menekan tombol "*Update*" dengan mengubah salah satu data pengguna	Melakukan update terhadap data pengguna kedalam *database*	Data pada *database* berhasil diperbaharui dan muncul pesan konfirmasi "*Update data berhasil*"	[√] Diterima [ ] Ditolak
Menekan tombol "Hapus"	Menghapus data pengguna dari dalam *database*	Data pengguna terhapus dari *database* dan muncul pesan konfirmasi "*Data berhasil dihapus*"	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Menekan tombol "*Create*" tanpa data masukan	Muncul pesan kesalahan "*Mohon maaf, Semua Input Form Harus Diisi !*"	Sistem menampilkan pesan kesalahan "*Mohon maaf, Semua Input Form Harus Diisi !*"	[√] Diterima [ ] Ditolak
Menekan tombol "*Update*" dengan mengosongkan salah ssatu *field* masukan	Muncul pesan kesalahan "*Mohon maaf, Semua Input Form Harus Diisi !*"	Sistem menampilkan pesan kesalahan "*Mohon maaf, Semua Input Form Harus Diisi !*"	[√] Diterima [ ] Ditolak

5) Kelola Data Guru

Kelola data Guru merupakan fungsional dari sistem untuk melakukan pengelolaan data Guru. Tabel 4.11 Berikut merupakan hasil pengujian fungsional kelola data pengguna.

Tabel 4.11 Hasil Pengujian Kelola Data Guru

Kasus dan Hasil Uji (Data Benar)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Mengakses halaman admin	Menampilkan data Guru dari *database*	Menampilkan data daftar Guru dari	[√] Diterima [ ] Ditolak

		database	
Menekan tombol "*Create*" dengan data masukan : Nama : "Rian Ariansyah", NIS : "101110800", Tempat Lahir : "Sumedang", Tanggal Lahir: "3 Desember 1991" Alamat : "komplek taman kopo indah I blok 2A" Telepon : "081222692904"	Menyimpan data Guru baru kedalam *database*	Data tersimpan kedalam *database* dan muncul pesan konfirmasi "*Data berhasil disimpan*"	[√] Diterima [ ] Ditolak
Menekan tombol "*Update*" dengan mengubah salah satu data pengguna	Melakukan update terhadap data Guru kedalam *database*	Data pada *database* berhasil diperbaharui dan muncul pesan konfirmasi "*Update data berhasil*"	[√] Diterima [ ] Ditolak
Menekan tombol "Hapus"	Menghapus data Guru dari dalam *database*	Data guru terhapus dari *database* dan muncul pesan konfirmasi "*Data berhasil dihapus*"	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Menekan tombol "*Create*" tanpa data masukan	Muncul pesan kesalahan "*Mohon maaf, Semua Input Form Harus Diisi !*"	Sistem menampilkan pesan kesalahan "*Mohon maaf, Semua Input Form Harus Diisi !*"	[√] Diterima [ ] Ditolak
Menekan tombol "*Update*" dengan mengosongkan salah ssatu *field* masukan	Muncul pesan kesalahan "*Mohon maaf, Semua Input Form Harus Diisi !*"	Sistem menampilkan pesan kesalahan "*Mohon maaf, Semua Input Form Harus Diisi !*"	[√] Diterima [ ] Ditolak

6) Kelola Data Kelas

Kelola data Kelas merupakan fungsional dari sistem untuk melakukan pengelolaan data Kelas. Tabel 4.12 Berikut merupakan hasil pengujian fungsional kelola data Kelas.

Tabel 4.12 Hasil Pengujian Kelola Data Kelas

Kasus dan Hasil Uji (Data Benar)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Mengakses halaman admin	Menampilkan data Kelas dari *database*	Menampilkan data daftar pengguna dari *database*	[√] Diterima [ ] Ditolak
Menekan tombol "*Create*"	Menyimpan data	Data tersimpan	[√] Diterima

dengan data masukan : Nama Kelas: TKJ-1", Wali Kelas : "Rifki Januar",	Kelas baru kedalam *database*	kedalam *database* dan muncul pesan konfirmasi "*Data berhasil disimpan*"	[ ] Ditolak
Menekan tombol "*Update*" dengan mengubah salah satu data pengguna	Melakukan update terhadap data Kelas kedalam *database*	Data kelas pada *database* berhasil diperbaharui dan muncul pesan konfirmasi "*Update data berhasil*"	[√] Diterima [ ] Ditolak
Menekan tombol "Hapus"	Menghapus data Kelas dari dalam *database*	Data kelas terhapus dari *database* dan muncul pesan konfirmasi "*Data berhasil dihapus*"	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Menekan tombol "*Create*" tanpa data masukan	Muncul pesan kesalahan "*Mohon maaf, Semua Input Form Harus Diisi !*"	Sistem menampilkan pesan kesalahan "*Mohon maaf, Semua Input Form Harus Diisi !*"	[√] Diterima [ ] Ditolak
Menekan tombol "*Update*" dengan mengosongkan salah ssatu *field* masukan	Muncul pesan kesalahan "*Mohon maaf, Semua Input Form Harus Diisi !*"	Sistem menampilkan pesan kesalahan "*Mohon maaf, Semua Input Form Harus Diisi !*"	[√] Diterima [ ] Ditolak

Aplikasi Absensi

7) Absensi Datang

Tabel 4.13 Hasil Pengujian Absensi Datang

Kasus dan Hasil Uji (Data Benar)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Masuk ke aplikasi absensi	Menampilkan aplikasi absensi	Sistem menampilkan aplikasi absensi	[√] Diterima [ ] Ditolak
Membaca kartu pelajar RFID yang sudah terdaftar	Membaca ID RFID dan memverifikasi data siswa kemudian mencatat jam masuk	Sistem membaca RFID dan memverifikasi data siswa dari *database* kemudian mencatat jam masuk ke dalam *datavase*	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Membaca kartu pelajar RFID yang belum terdaftar	Membaca ID RFID dan memverifikasidata siswa kemudian menampilkan pesan	Sistem membaca RFID dan memverifikasi data siswa dari *database* kemudian menampilkan pesan "Kartu Pelajar yang Masukan Tidak Terdaftar"	[√] Diterima [ ] Ditolak

8) Absensi Keluar

Tabel 4.14 Hasil Pengujian Absensi Keluar

Kasus dan Hasil Uji (Data Benar)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Masuk ke aplikasi absensi	Menampilkan aplikasi absensi	Sistem menampilkan aplikasi absensi	[√] Diterima [ ] Ditolak
Membaca kartu pelajar RFID yang sudah terdaftar	Membaca ID RFID dan memverifikasidata siswa kemudian mencatat jam keluar	Sistem membaca RFID dan memverifikasi data siswa dari *database* kemudian mencatat jam keluar ke dalam *datavase*	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Membaca kartu pelajar RFID yang belum terdaftar	Membaca ID RFID dan memverifikasidata siswa kemudian menampilkan pesan	Sistem membaca RFID dan memverifikasi data siswa dari *database* kemudian menampilkan pesan "Kartu Pelajar yang Masukan Tidak Terdaftar"	[√] Diterima [ ] Ditolak

9) Mengisi Keterangan

Tabel 4.15 Hasil Pengujian Mengisi Keterangan

Kasus dan Hasil Uji (Data Benar)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Masuk ke aplikasi absensi	Menampilkan aplikasi absensi	Sistem menampilkan aplikasi absensi	[√] Diterima [ ] Ditolak
Memasukan keterangan absensi siswa	Guru memasukan keterangan tidak hadir seperti "sakit", "izin", dan "tanpa keterangan"	Sistem menampilkan pilihan keterangan tidak hadir "sakit", "izin", dan "tanpa keterangan", guru memilih kemudian *update* data absensi ke dalam database	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Memasukan keterangan tidak hadir yang salah	Sistem akan memberitahu keterangan tidak hadir salah	Sistem berhasil update database absen	[ ] Diterima [√] Ditolak

Aplikasi Perpustakaan

10) Pengunjung Perpustakaan

Tabel 4.16 Hasil Pengujian Aplikasi Perpustakaan

Kasus dan Hasil Uji (Data Benar)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Masuk ke aplikasi perpustakaan	Menampilkan aplikasi perpustakaan	Sistem menampilkan aplikasi perpustakaan	[√] Diterima [ ] Ditolak
Membaca kartu pelajar RFID yang sudah terdaftar	Membaca ID RFID, memverifikasidata siswa dan data *smartcard* kemudian mencatat jam masuk	Sistem membaca RFID dan memverifikasi data siswa dan data *smartcard* dari *database* kemudian mencatat jam masuk ke dalam *datavase*	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Membaca kartu pelajar RFID yang belum terdaftar	Membaca ID RFID dan memverifikasidata siswa dan data *smartcard* kemudian menampilkan pesan	Sistem membaca RFID dan memverifikasi data siswa dari *database* kemudian menampilkan menampilkan pesan "Kartu Pelajar yang Masukan Tidak Terdaftar"	[√] Diterima [ ] Ditolak

11) Peminjaman Buku

Tabel 4.17 Hasil Pengujian Peminjaman Buku

Kasus dan Hasil Uji (Data Benar)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Masuk ke aplikasi perpustakaan	Menampilkan aplikasi perpustakaan	Sistem menampilkan aplikasi perpustakaan	[√] Diterima [ ] Ditolak
Memasukan judul buku dan lama pinjaman, kemudian membaca RFID	Guru dapat memasukan judul buku dan lama pinjaman, membaca ID RFID dan memverifikasi data siswa dan data smartcard kemudian mencatat detail peminjaman buku	Sistem memasukan judul buku dan lama pinjaman, membaca ID RFID dan memverifikasidata siswa dan data smartcard kemudian mencatat detail peminjaman buku kemudian *update* data absensi ke dalam database	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Membaca kartu pelajar RFID yang belum terdaftar	Membaca ID RFID dan memverifikasidata siswa dan data *smartcard* kemudian menampilkan pesan	Sistem membaca RFID dan memverifikasi data siswa dari *database* kemudian menampilkan menampilkan pesan "Kartu Pelajar yang Masukan Tidak Terdaftar"	[√] Diterima [ ] Ditolak

12) Pengembalian Buku

Tabel 4.18 Hasil Pengujian Pegembalian Buku

Kasus dan Hasil Uji (Data Benar)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Masuk ke aplikasi perpustakaan	Menampilkan aplikasi perpustakaan	Sistem menampilkan aplikasi perpustakaan	[√] Diterima [ ] Ditolak
Membaca kartu pelajar RFID yang sudah terdaftar	Membaca ID RFID dan memverifikasi data siswa dan data smartcard kemudian menampilkan detail peminjaman buku	istem membaca RFID dan memverifikasi data siswa dari *database* kemudian menampilkan data peminjaman dari dalam *datavase*	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Membaca kartu pelajar RFID yang belum terdaftar	Membaca ID RFID dan memverifikasidata siswa dan data *smartcard* kemudian menampilkan pesan	Sistem membaca RFID dan memverifikasi data siswa dari *database* kemudian menampilkan menampilkan pesan "Kartu Pelajar yang Masukan Tidak Terdaftar"	[√] Diterima [ ] Ditolak

Aplikasi Parkir

13) Parkir Masuk

Tabel 4.19 Hasil Pengujian Parkir Masuk

Kasus dan Hasil Uji (Data Benar)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Masuk ke aplikasi parkir	Menampilkan aplikasi parkir	Sistem menampilkan aplikasi parkir	[√] Diterima [ ] Ditolak
Membaca kartu pelajar RFID yang sudah terdaftar	Membaca ID RFID, memverifikasi data siswa dan data *smartcard* kemudian mencatat jam masuk	Sistem membaca RFID dan memverifikasi data siswa dan data *smartcard* dari *database* kemudian mencatat jam masuk ke dalam *datavase*	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Membaca kartu pelajar RFID yang belum terdaftar	Membaca ID RFID dan memverifikasidata siswa dan data *smartcard* kemudian menampilkan pesan	Sistem membaca RFID dan memverifikasi data siswa dan data *smartcard* dari *database* kemudian menampilkan pesan "Kartu Pelajar yang Masukan Tidak Terdaftar"	[√] Diterima [ ] Ditolak
Melakukan prosedur parkir masuk saat belum melakukan	Membaca ID RFID dan memverifikasi data siswa dan data	Sistem membaca RFID dan memverifikasi data siswa dan data *smartcard*	[√] Diterima [ ] Ditolak

prosedur parkir keluar	*smartcard* kemudian menampilkan pesan	dari *database* kemudian menampilkan pesan "tidak bias parkir karena belum melakukan parkir keluar"	

14) Parkir Keluar

Tabel 4.20 Hasil Pengujian Parkir Keluar

Kasus dan Hasil Uji (Data Benar)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Masuk ke aplikasi Parkir	Menampilkan aplikasi parkir	Sistem menampilkan aplikasi parkir	[√] Diterima [ ] Ditolak
Membaca kartu pelajar RFID yang sudah terdaftar	Membaca ID RFID dan memverifikasi data siswa dan data smartcard kemudian mencatat jam keluar	Sistem membaca RFID dan memverifikasi data siswa dari *database* kemudian mencatat jam keluar ke dalam *datavase*	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Membaca kartu pelajar RFID yang belum terdaftar	Membaca ID RFID dan memverifikasidata siswa dan data *smartcard* kemudian menampilkan pesan	Sistem membaca RFID dan memverifikasi data siswa dan data *smartcard* dari *database* kemudian menampilkan pesan "Kartu Pelajar yang Masukan Tidak Terdaftar"	[√] Diterima [ ] Ditolak

4.2.1.3 Kesimpulan Pengujian Alpha

Berdasarkan hasil pengujian *Alpha* yang telah dilakukan, maka dapat disimpulkan bahwa secara fungsional sistem yang memanfaatkan RFID sebagai kartu pelajar pintar dalam penelitian ini telah berjalan sesuai dengan yang diharapkan. Sistem dapat melakukan fungsional dari dari mulai proses pendaftaran kartu pelajar sampai dengan prosedur-prosedur lainnya.

4.2.2 Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan untuk mengetahui sejauh mana kegunaan aplikasi yang dibangun, apakah telah memenuhi harapan atau belum. Untuk itu dalam pengujian beta ini akan dilakukan penelitian terhadap sejumlah responden atau pengguna sistem. Pengujian beta dilakukan dengan memberikan format kuesioner terhadap 50 siswa.

4.2.2.1 Kuesioner Terhadap Siswa SMKN 1 Katapang

Perhitungan untuk kuesioner menggunakan skala likert, dimana setiap pertanyaan mempunyai bobot nilai. Adapun bobot jawaban pada kuesioner adalah sebagai berikut:

Tabel 4.21 Bobot Jawaban Kuisioner

Kategori Jawaban	Skor
Sangat Setuju	5
Setuju	4
Biasa Saja	3
Tidak Setuju	2
Sangat Tidak Setuju	1

Untuk mencari persentase dari masing-masing jawaban kuesioner digunakan rumus skala likert sebagai berikut :

$$\boldsymbol{P = \frac{S}{Skor\ Ideal} \times 100\%}$$

Keterangan rumus mencari nilai presentase dari jawaban kuesioner dapat dilihat pada tabel 4.22

Tabel 4.22 Rumus Mencari Nilai Presentase Kuisioner

Nama	Keterangan
P	Nilai persentase yang dicari
S	Jumlah frekuensi dikalikan dengan skor yang ditetapkan jawaban
Skor Ideal	Nilai tertinggi dikalikan dengan jumlah sampel
Nilai Tertinggi	5
Populasi	50
Nilai Skor Ideal	5 x 50 = 250

Hasil Pengujian

Hasil dari pengujian yang dilakukan dengan menggunakan kuesioner terhadap siswa SMKN 1 Katapang, didapat hasil pengumpulan data dari masing-masing jawaban dari setiap responden, dapat dilihat pada tabel 4.23

Tabel 4.23 Hasil Kuisioner

No	Pertanyaan	Jawaban					Total
		SS	S	BS	TS	STS	
1	Apakah sistem kartu pelajar pintar yang dibangun mudah untuk digunakan?	26	8	6	0	0	50
2	Apakah dengan kartu pintar membantu anda dalam melakukan parkir?	5	1	2	8	4	50
3	Apakah dengan kartu pintar membantu anda dalam melakukan peminjaman diperpustakaan?	17	6	5	3	0	50
4	Apakah dengan kartu pintar memudahkan anda dalam melakukan absensi?	15	5	6	3	1	50
5	Apakah sistem kartu pintar yang sudah di terapkan sudah sesuai dengan penggunaan yang ada?	14	2	2	2	0	50
6	Apakah fitur dalam sistem kartu pintar ini sudah lengkap?	21	0	5	4	0	50

Untuk mengetahui interpretasi skor hasil perhitungan dapat dilihat pada tabel 4.24

Tabel 4.24 Interpretasi Skor Hasil Perhitungan

Nilai Persentase	Keterangan
0 % – 20 %	Sangat Tidak Setuju
21 % – 40 %	Tidak Setuju
41 % – 60 %	Biasa Saja
61 % – 80 %	Setuju
81 % – 100 %	Sangat Setuju

Berikut adalah tabel-tabel hasil perhitungan jawaban dari masing-masing pertanyaan pada kuisioner yang telah disebar kepada seluruh siswa kelas X

Perhitungan pertanyaan ke-1

Berikut adalah hasil perhitungan pernyataan ke-1, Apakah sistem kartu pelajar pintar yang dibangun mudah untuk digunakan?

Tabel 4.25 Perhitungan Pertanyan Ke-1

Keterangan	Skor	Responden	Jumlah Skor
Sangat Setuju	5	26	130
Setuju	4	18	72

Keterangan	Skor	Responden	Jumlah Skor
Biasa Saja	3	6	18
Tidak Setuju	2	0	0
Sangat Tidak Setuju	1	0	0
Jumlah		50	220

Y = (220/250) x 100% = 88%

Dari hasil perhitungan di atas, dapat diketahui bahwa rata-rata dari responden sangat setuju dengan pernyataan, Apakah Aplikasi Pembelajaran Bahasa Inggris ini mudah untuk Digunakan?

Perhitungan pertanyaan ke-2

Berikut adalah hasil perhitungan pernyataan ke-2, Apakah dengan kartu pintar membantu anda dalam melakukan parkir?

Tabel 4.26 Perhitungan Pertanyan Ke-2

Keterangan	Skor	Responden	Jumlah Skor
Sangat Setuju	5	5	25
Setuju	4	21	84
Biasa Saja	3	12	36
Tidak Setuju	2	8	16
Sangat Tidak Setuju	1	4	4
Jumlah		50	165

Y = (165/250) x 100% = 66%

Dari hasil perhitungan di atas, dapat diketahui bahwa rata-rata dari responden setuju dengan pernyataan?

Perhitungan pertanyaan ke-3

Berikut adalah hasil perhitungan pernyataan ke-3, Apakah dengan kartu pintar membantu anda dalam melakukan peminjaman di perpustakaan?

Tabel 4.27 Perhitungan Pertanyan Ke-3

Keterangan	Skor	Responden	Jumlah Skor
Sangat Setuju	5	17	85
Setuju	4	15	60

Keterangan	Skor	Responden	Jumlah Skor
Biasa Saja	3	15	45
Tidak Setuju	2	3	6
Sangat Tidak Setuju	1	0	0
Jumlah		50	196

Y = (196/250) x 100% = 78.4%

Dari hasil perhitungan di atas, dapat diketahui bahwa rata-rata dari responden sangat setuju dengan pernyataan?

Perhitungan pertanyaan ke-4

Berikut adalah hasil perhitungan pernyataan ke-4, Apakah dengan kartu pintar memudahkan anda dalam melakukan absensi?

Tabel 4.28 Perhitungan Pertanyan Ke-4

Keterangan	Skor	Responden	Jumlah Skor
Sangat Setuju	5	15	75
Setuju	4	15	60
Biasa Saja	3	16	48
Tidak Setuju	2	3	6
Sangat Tidak Setuju	1	1	1
Jumlah		50	190

Y = (190/250) x 100% = 76%

Dari hasil perhitungan di atas, dapat diketahui bahwa rata-rata dari responden sangat setuju dengan pernyataan?

Perhitungan pertanyaan ke-5

Berikut adalah hasil perhitungan pernyataan ke-5, Apakah sistem kartu pintar yang sudah di terapkan sudah sesuai dengan penggunaan yang ada?

Tabel 4.29 Perhitungan Pertanyan Ke-5

Keterangan	Skor	Responden	Jumlah Skor
Sangat Setuju	5	14	70
Setuju	4	22	88
Biasa Saja	3	12	36

Keterangan	Skor	Responden	Jumlah Skor
Tidak Setuju	2	2	4
Sangat Tidak Setuju	1	0	0
Jumlah		50	198

Y = (198/250) x 100% = 79.2%

Dari hasil perhitungan di atas, dapat diketahui bahwa rata-rata dari responden sangat setuju dengan pernyataan,

Perhitungan pertanyaan ke-6

Berikut adalah hasil perhitungan pernyataan ke-6, Apakah fitur dalam sistem kartu pintar ini sudah lengkap?

Tabel 4.30 Perhitungan Pertanyan Ke-6

Keterangan	Skor	Responden	Jumlah Skor
Sangat Setuju	5	21	105
Setuju	4	10	40
Biasa Saja	3	15	45
Tidak Setuju	2	4	8
Sangat Tidak Setuju	1	0	0
Jumlah		50	198

Y = (198/250) x 100% = 79.2%

Dari hasil perhitungan di atas, dapat diketahui bahwa rata-rata dari responden sangat setuju dengan pernyataan?

Kesimpulan Hasil Kuisioner

Setiap pertanyaan yang diajukan pada kuesioner yang dibagikan kepada 50 siswa memiliki jumlah persentase yang berbeda. Berdasarkan hasil presentase yang didapatkan melalui perhitungan kuisioner setiap pernyataan maka dapat dihitung presentase kuisioner secara keseluruhan dengan rumus sebagai berikut:

PK = Σ presentase pertanyaan sample / Σ presentase keseluruhan

PK = (88+66+78.4+76+79.2+79.2) / 6

PK = 466.8 / 6 = 77.8

Maka dapat disimpulkan bahwa hasil perhitungan kuesioner dari 50 siswa menyatakan 77.8% responden menyetujui pertanyaan-pertanyaan yang diberikan.